APRENDE A PROGRAMAR

GRÁFICOS



Ediciones Generales Anaya

PARA SPECTRUM Y COMMODORE

Diseño: Cooper West
Edición: James McCarter
Programas: Steve Rodgers
Marcus Milton
Ilustraciones: Lionel Jeans
Rob Shone

Traducción: Pedro Garre del Olmo

Agradecemos su colaboración a
J. David García Rodríguez

Título original: GRAPHICS

Villafranca, 22. 28028 Madrid
ISBN.: 84-7525-315-6
Depósito legal: M. 39.271/1985
Impreso por: Melsa. Ctra. de Fuenlabrada a Pinto, km. 21,800. Pinto (Madrid)
Impreso en España - Printed in Spain


APRENDE A PROGRAMAR

GRÁFICOS



Mike Duck

Ediciones Generales Anaya



Introducción

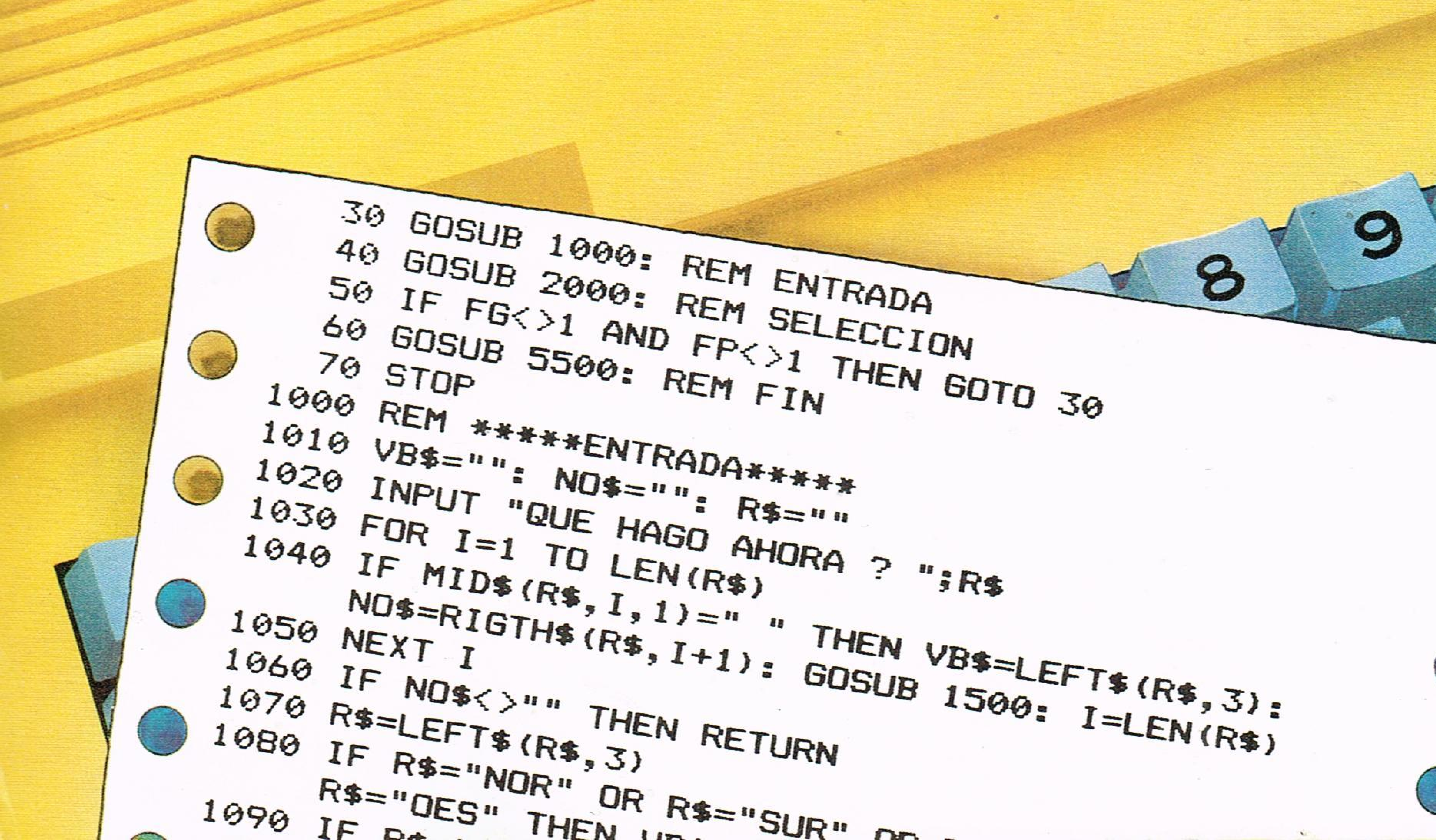
Una de las cosas más interesantes que pueden hacer los ordenadores es imprimir gráficos en la pantalla del televisor. La mayoría de los programas de juegos te permiten controlar lo que ocurre en la pantalla, tanto si se trata de una nave espacial que se mueve para eludir el ataque enemigo, como si se intenta conducir un coche de carreras por un peligroso circuito de Grand Prix. Los programas que crean estos efectos son muy complicados, pero, por medio de instrucciones sencillas, puedes crear por ti mismo juegos muy interesantes.

El programa que se da en este libro desarrolla el juego del ahorcado. En este juego, el ordenador elige una palabra misteriosa y el jugador debe adivinarla letra a letra. Para ello sólo tienes diez intentos y, por cada letra que falles, el ordenador dibujará una parte del gráfico del ahorcado.

El programa está dividido en unidades lógicas, para que puedas entender claramente lo que hace cada sección. Dando a los programas una estructura lógica paso a paso, no sólo los haces más fáciles de escribir, sino que serán más sencillos de entender para cualquiera que utilice este programa en el futuro. La primera parte del libro trata sobre el programa principal, que controla todo el juego; en la segunda se explican los gráficos que aparecerán en la pantalla.

Contenido

Introducción a los gráficos	8
Color	9
La mejor manera de programar	10
1 EL PROGRAMA PRINCIPAL	11
COMMODORE	12
SPECTRUM	19
2 GRÁFICOS	27
COMMODORE	28
SPECTRUM	34
Listado de los programas	40



Introducción a los gráficos

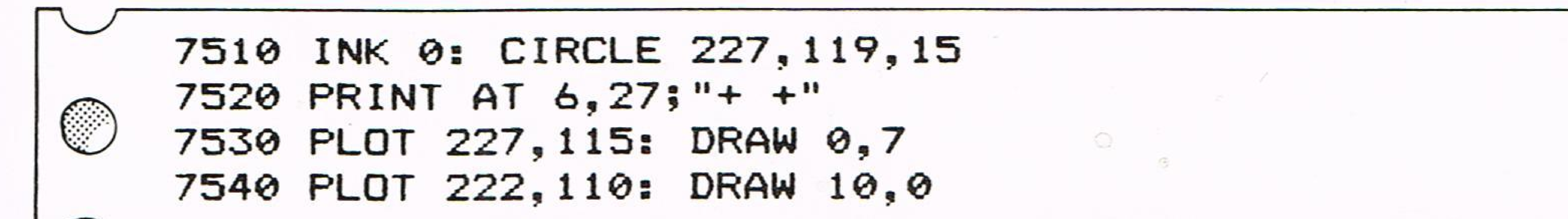
Las instrucciones gráficas usadas en BASIC varían considerablemente de un modelo de ordenador a otro.
A diferencia de otros ordenadores, el Spectrum sólo posee un método para realizar gráficos. Los gráficos se dibujan en la pantalla de la misma forma que se dibujarían en un papel.
Para ello usa varias instrucciones gráficas seguidas de dos números, el primero indicando la coordenada horizontal y el segundo la vertical. No olvides los dos puntos (:) para separar cada instrucción.
El Commodore 64 también puede colocar gráficos en la pantalla, utilizando una serie de instrucciones **PRINT.**
Teclea las instrucciones que aparecen abajo y ejecútalas para ver los resultados.

Commodore

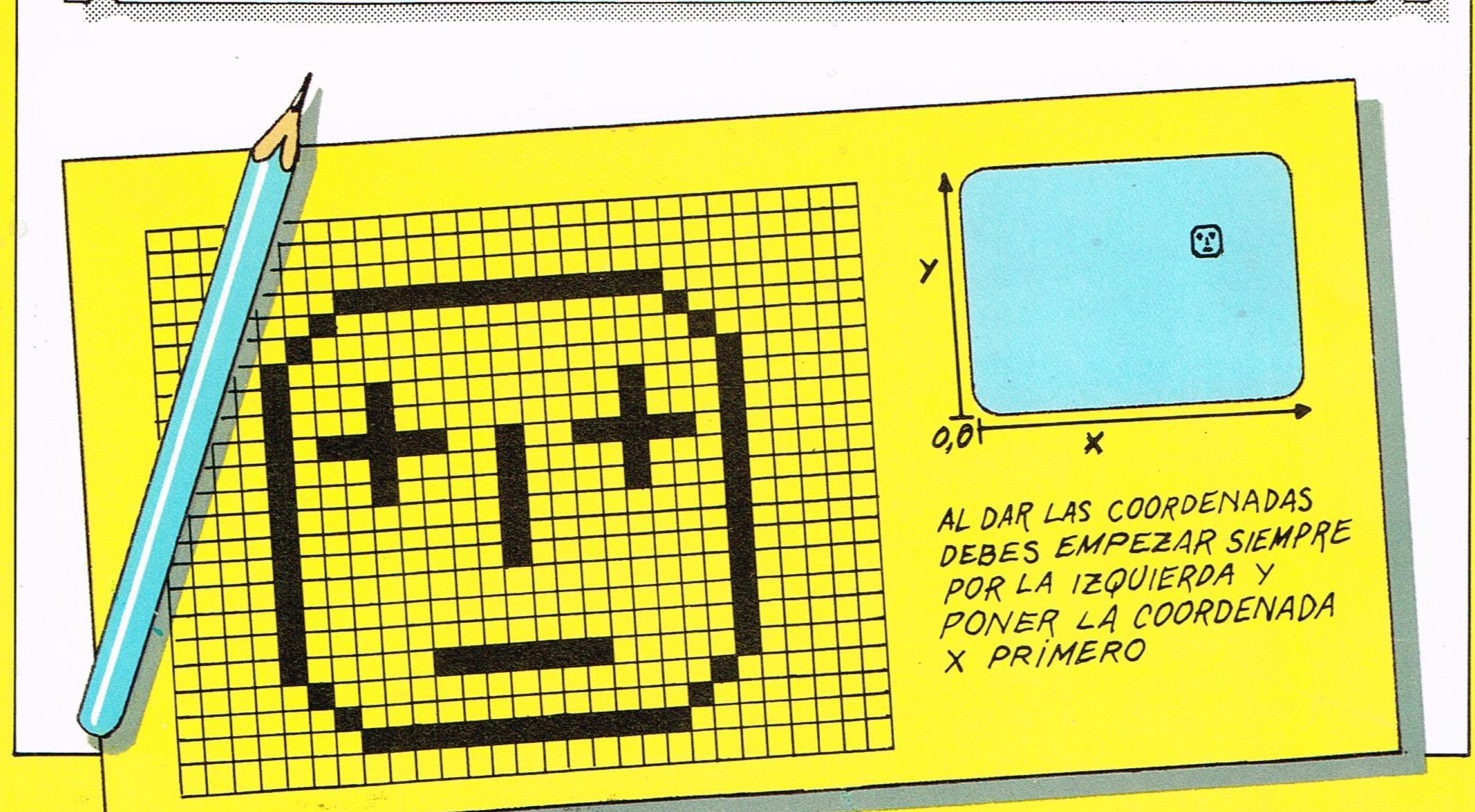


```
7510 PRINT"▒▒▒▒▒▒▒▒"
7520 PRINT SPC(27)" ╭─┴─╮"
7530 FORR=0TO2:PRINTSPC(27)"|    |":NEXT
7540 PRINT SPC(27)" ╰─┬─╯TTTT"
7550 PRINT SPC(28)"+ +": PRINT SPC(29)":":
     PRINT SPC(28)"───"
```

Spectrum



```
7510 INK 0: CIRCLE 227,119,15
7520 PRINT AT 6,27;"+ +"
7530 PLOT 227,115: DRAW 0,7
7540 PLOT 222,110: DRAW 10,0
```



Color

En cada modelo de ordenador se utiliza un conjunto de instrucciones distinto para la obtención de colores. Los gráficos y textos del Spectrum se colorean por medio de la instrucción **INK**, seguida de un número entre uno y siete. El fondo del papel se obtiene con la instrucción **PAPER**, seguida de un número del mismo rango anterior. En el Commodore 64 los gráficos y textos se colorean con la tecla **C=** ó **CTRL**, acompañados de un número entre uno y ocho. Cada control de color imprimirá un único código gráfico, que no será visible cuando el programa se ejecute. El color del papel en el Commodore es azul.
Prueba las instrucciones para el Spectrum que tienes más abajo y experimenta con los valores de las tablas.

Códigos de control del Commodore

- ♥ = BORRAR PANTALLA
- S = POSICION INICIAL CURSOR
- Q = CURSOR HACIA ABAJO
- ● = CURSOR ARRIBA
-] = CURSOR DERECHA
- | = CURSOR IZQUIERDA
- R = MODO INVERSO
- ▁ = MODO NORMAL

Spectrum



```
10 PAPER 6: INK 2: CLS
20 PRINT AT 5,7;"ESTAS LETRAS SON ROJAS"
30 INK 5: PLOT 20,20: DRAW 100,100: DRAW 100,-100
```



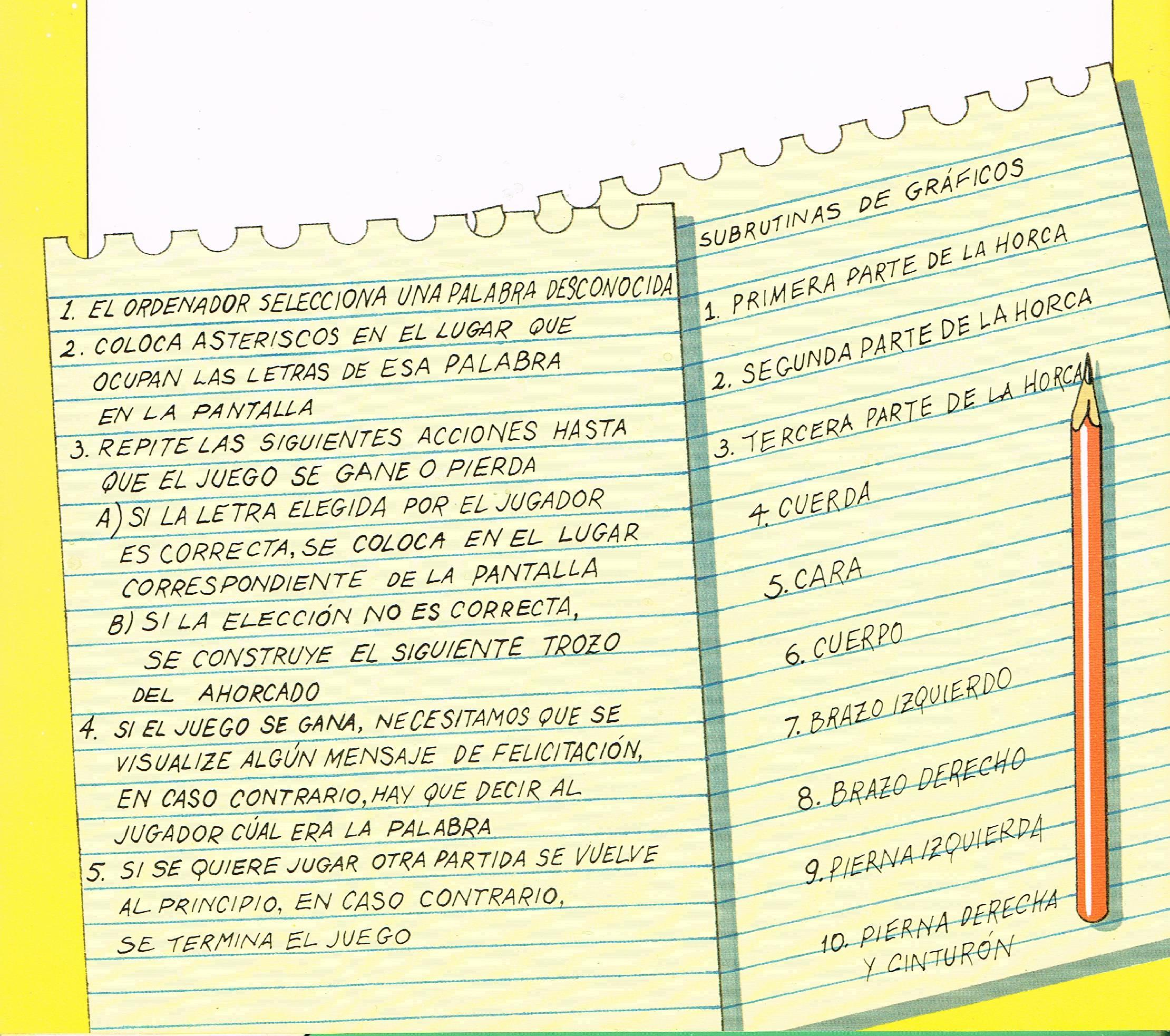
Los colores de tinta y papel en el Spectrum y en el Commodore

COLOR DE TINTA	
SPECTRUM	COMMODORE
HCOLOR0 HCOLOR 4	CTRL 1
	CTRL 4
HCOLOR 5	CTRL 3
	CTRL 5
HCOLOR 1	CTRL 6
HCOLOR 6 HCOLOR 2	CTRL 7
	CTRL 8
HCOLOR 3 HCOLOR 7	CTRL 2

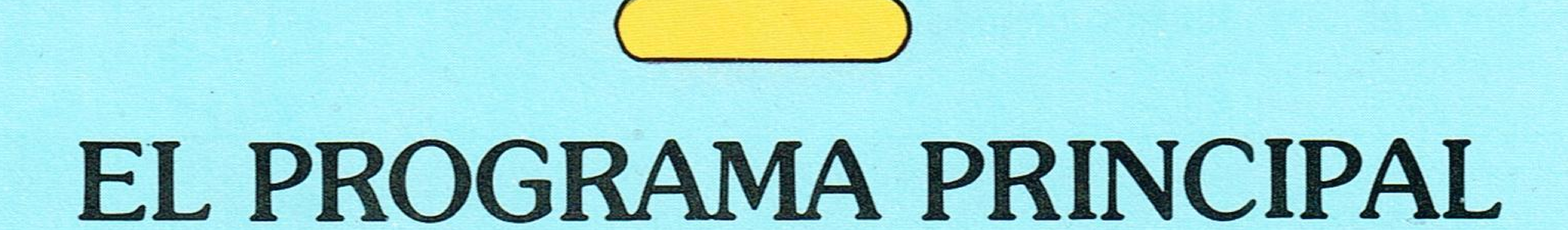
OTROS COLORES DEL COMMODORE		
NARANJA	C=	1
MARRON	C=	2
ROJIZO	C=	3
GRIS OSCURO	C=	4
GRIS	C=	5
VERDE CLARO	C=	6
AZUL CLARO	C=	7
GRIS CLARO	C=	8

La mejor manera de programar

La primera etapa en la escritura de un programa es dividir la tarea que va a realizar el ordenador, en pasos lógicos. Cada uno de estos pasos se puede hacer en unidades separadas del programa, llamadas *subrutinas* en el BASIC (lenguaje utilizado por el Spectrum y el Commodore). El programa principal llamará y ejecutará cada una de estas subrutinas. Una de las ventajas de escribir los programas de esta manera es que se ve claramente cómo funcionan, y que se puede utilizar la misma subrutina más de una vez dentro del programa. Así, en el programa de este libro, los gráficos que se crean en la sección de gráficos se utilizan para formar pantallas atractivas cuando se gana, se pierde o se empieza el juego. El uso de subrutinas también hace que el programa principal sea mucho más simple; de hecho, el programa principal, con las llamadas a las subrutinas, ocupa unas diez líneas de programa.



1



EL PROGRAMA PRINCIPAL

En este apartado se explican las subrutinas que describen el juego del ahorcado. Primero se da el programa principal, con todos los pasos colocados en su orden lógico. A continuación el programa dice al ordenador que elija una palabra aleatoria de la lista de datos que él tiene, y que escriba la siguiente pantalla. Tras ello, se pide al jugador que elija letras, y estos intentos se comparan con las letras de la palabra misteriosa.

<u>Letras escogidas</u>

La palabra
del ordenador es:

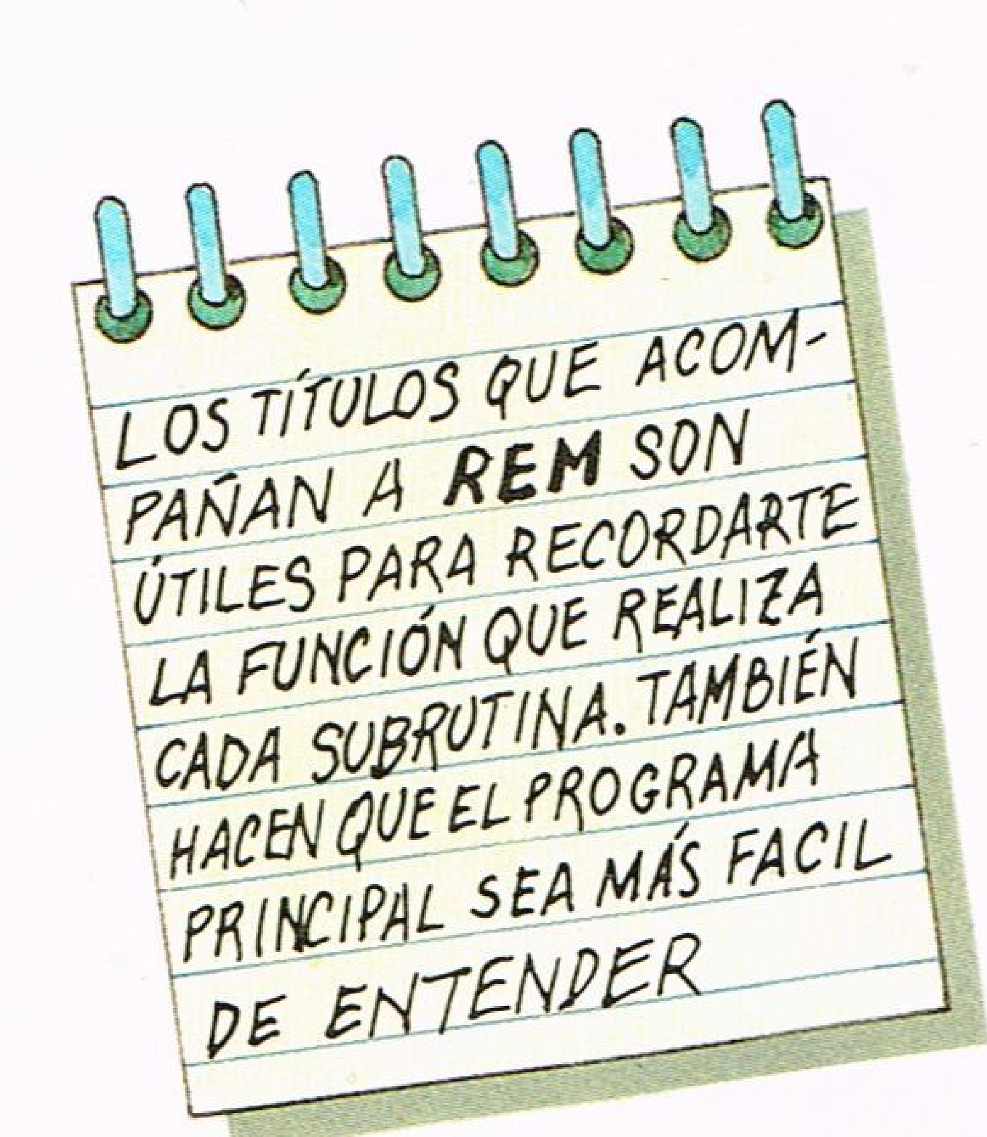
* * * * * * * *

Por favor, escoge una letra

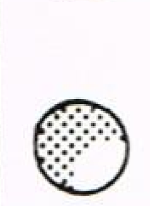
El programa principal del juego del ahorcado es muy parecido al esquema dado en la introducción. Cada paso lógico del programa se realiza en una subrutina. El programa principal está completo cuando se disponen las subrutinas en su orden lógico y se incluyen las pruebas necesarias para comprobar si el juego ha sido ganado o perdido.

```
 10 PRINT"♥"
 20 GOSUB 1000:REM SELECCIONA PALABRA
 30 GOSUB 2000:REM TITULO
 40 IF GA=1 OR PI=1THEN 80
 50 GOSUB 3000:REM ELIGE LETRA
 60 IF CO=0 THEN GOSUB 7000: REM HOMBRE AHORCADO
 70 GOTO 40
 80 IF GA =1 THEN GOSUB 4000: REM GANAR
 90 IF PI =1 THEN GOSUB 5000: REM PERDER
100 GOSUB 6000:REM OTRA PARTIDA
110 STOP
```

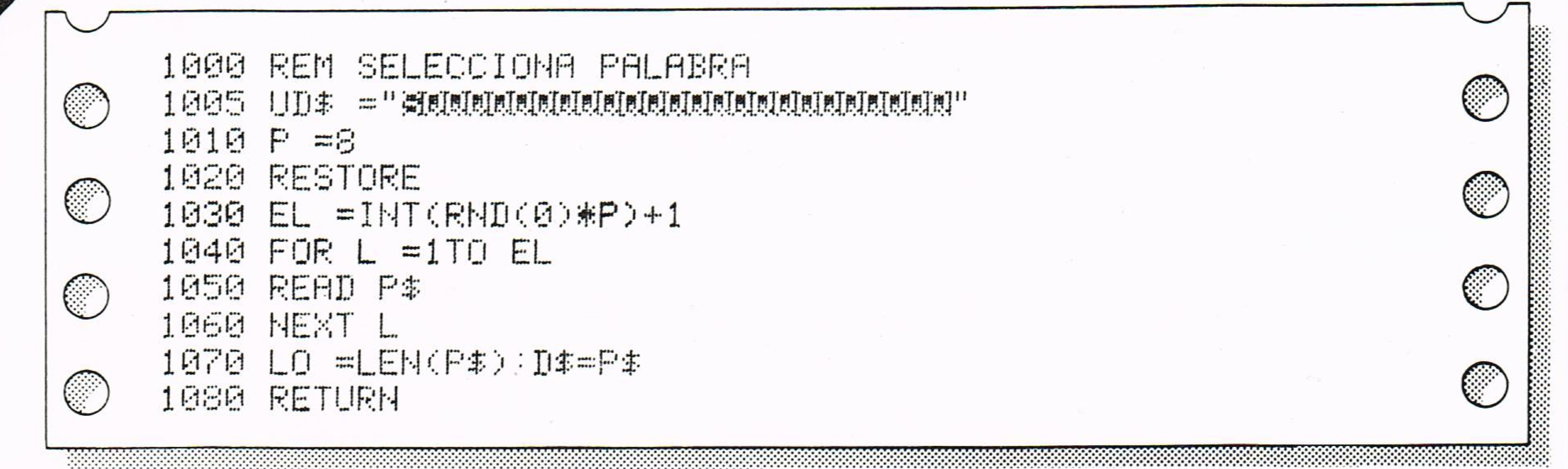
Observa el programa principal con más detalle. Notarás que comienza yendo a una subrutina que pide al ordenador que elija aleatoriamente una palabra de las almacenadas en el **DATA,** al final del programa. Después, al comienzo del juego, aparece en la pantalla la cabecera o título. La línea 40 comprueba si se ha ganado o perdido el juego (esto sólo se produce después de realizar un intento). La línea 70 devuelve el control a la 40 para que se pueda realizar otro intento. La llamada a la subrutina *elige letra* de la línea 50, pide al jugador que haga otro intento. En la línea 60, el programa va a la rutina *ahorcado*, que contiene los gráficos del juego. El valor de **CO** se pone a cero si se ha realizado un intento fallido en la rutina *elige letra*. Finalmente, hay rutinas que se ejecutan si el juego se ha ganado o perdido; otras preguntan al jugador si quiere jugar otra vez. En once líneas tienes descrito todo el juego.



Ya sólo nos falta escribir las rutinas (ahora es cuando empieza el trabajo duro). Antes de empezar, añade en el **DATA** la relación que damos abajo, para que puedas probar el programa a la vez que lo vas escribiendo. El ordenador elegirá una palabra aleatoriamente de esta lista en el transcurso del programa. Por supuesto, no es obligatorio utilizar las palabras que aquí te damos, pero por el momento es mejor que el número de palabras no sobrepase las ocho; podrás añadir más palabras cuando el programa esté acabado.

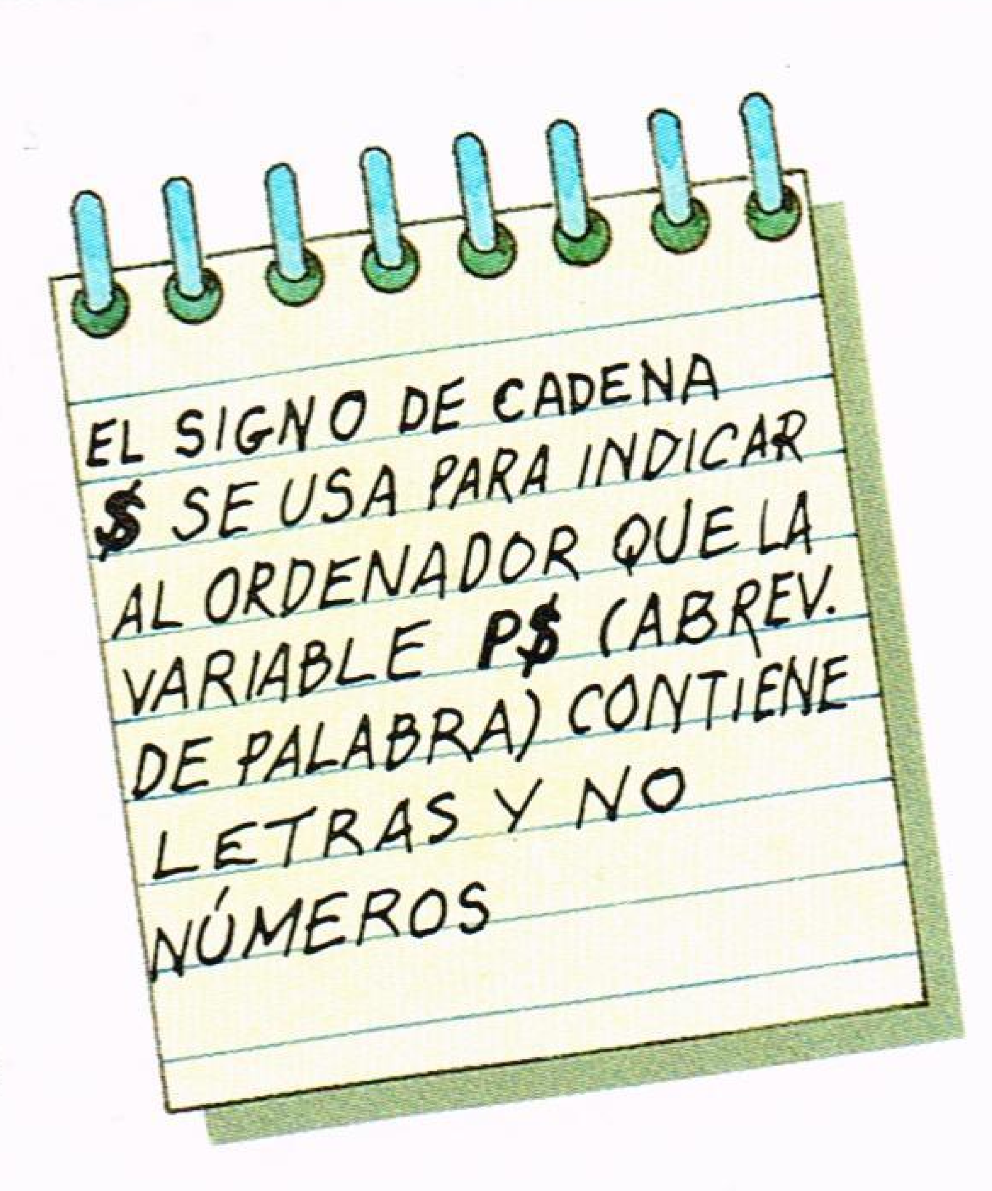


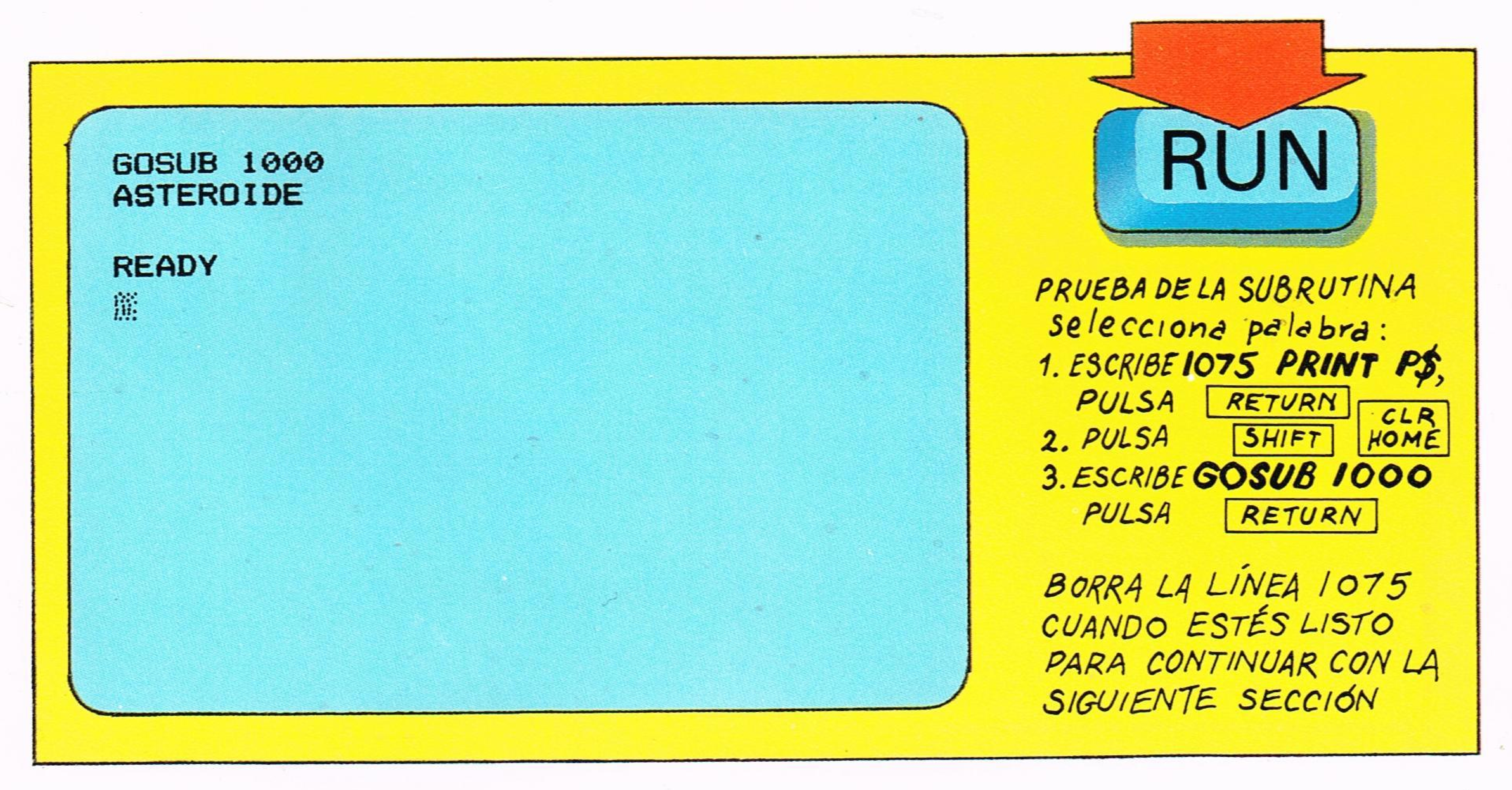
```
9000 DATA MANZANA,ANGULO,ALFABETO,ANTIGUO
9010 DATA ALUBIAS,LIBRO,BUFALO,PLAYA
```



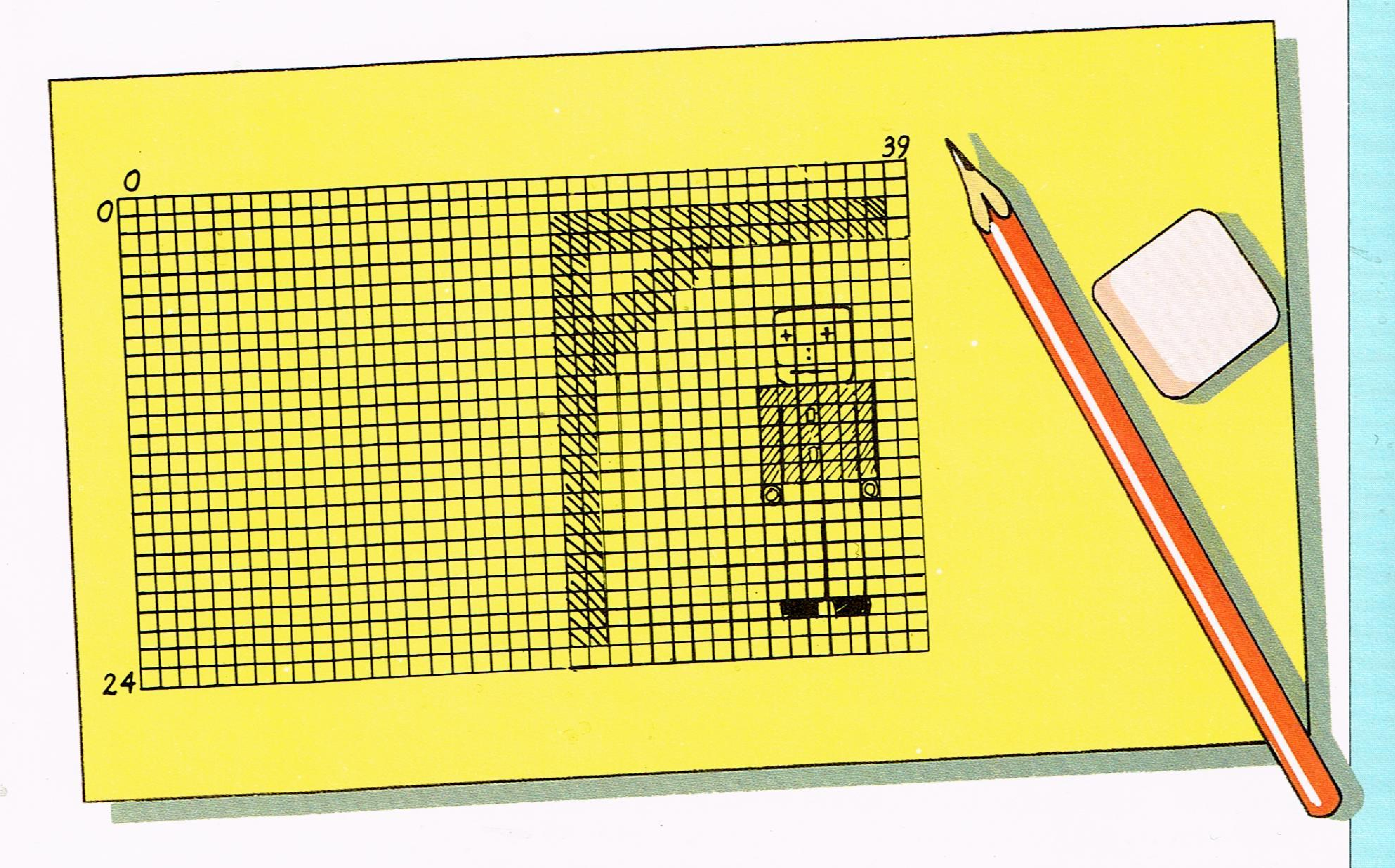
```
1000 REM SELECCIONA PALABRA
1005 UD$ ="[caracteres gráficos]"
1010 P =8
1020 RESTORE
1030 EL =INT(RND(0)*P)+1
1040 FOR L =1TO EL
1050 READ P$
1060 NEXT L
1070 LO =LEN(P$):D$=P$
1080 RETURN
```

La primera subrutina hace que el ordenador elija una palabra, aleatoriamente, de la lista almacenada en el **DATA**. En la línea 1010 se le da a **P** un valor igual a número de palabras que hay en el **DATA.** La instrucción **RESTORE** hace que el ordenador lea la lista de datos desde el principio cada vez que tiene que elegir una palabra. La línea 1030 genera un número aleatorio entre 0 y **P.** El Commodore 64 genera números aleatorios entre 0 y 1; el número generado lo multiplicaremos por el número de elementos del **DATA**, y el número conseguido lo pasaremos como parámetro de la instrucción **INT** para conseguir el número final. A este número le añadiremos 1 para que no pueda ser cero. El bucle **FOR...NEXT,** de las líneas 1040 a 1060, consigue que el ordenador lea en el **DATA** hasta el punto indicado por el número aleatorio. Por ejemplo, si el número aleatorio generado fuese cuatro, el ordenador leería hasta la cuarta palabra. Esta palabra se almacenará en **P$.** La línea 1070 encontrará la longitud de la palabra misteriosa y almacenará esta palabra en **D$. D$** se usa para escribir en pantalla la palabra misteriosa, en el caso de que se pierda el juego. Finalmente, **RETURN** envía, de nuevo, al programa principal.





La siguiente subrutina escribe la pantalla de presentación; usará alguno de los gráficos creados por la subrutina *ahorcado*. Estos aparecerán en el lado derecho de la pantalla; tenlo en cuenta cuando decidas dónde van a ir los rótulos en la pantalla.

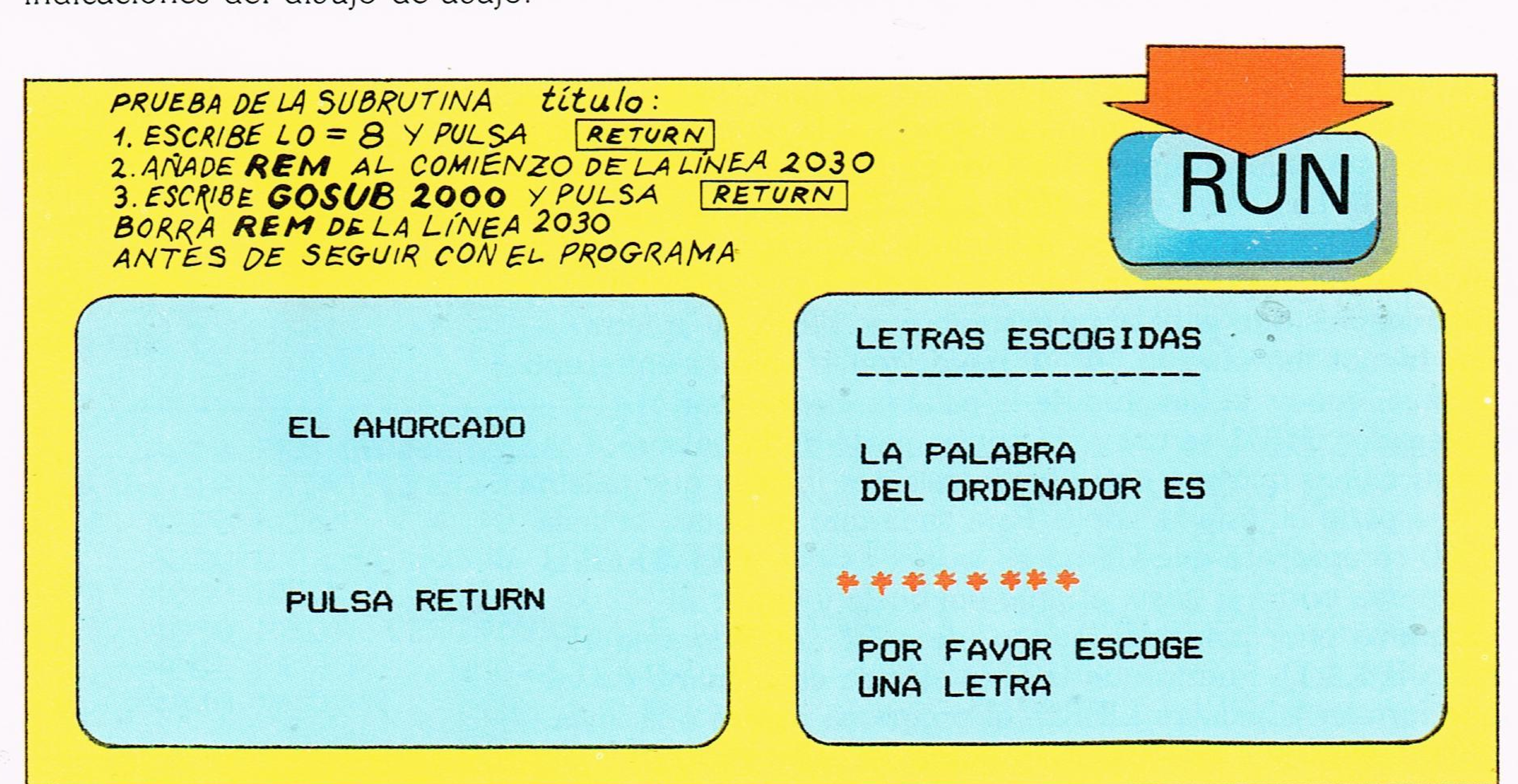


La rutina de presentación tiene dos partes. La primera, dada abajo, proporciona los rótulos de la pantalla inicial, y la segunda prepara la pantalla para el inicio del juego. La línea 2010 borra la pantalla. Las líneas de la 2020 a la 2040 mandan el programa a la sección apropiada de gráficos en la subrutina *ahorcado*, etapas de la cuatro a la diez. (Esto se explicará más adelante, en otra parte del libro.) En resumen, se irá formando la imagen del ahorcado en diez etapas, una por cada intento fallido o vida perdida. La instrucción **PRINT** coloca el rótulo «EL AHORCADO» en la línea décima de la pantalla, a doce columnas del margen izquierdo. El jugador es entonces invitado a pulsar **RETURN** para empezar el juego.

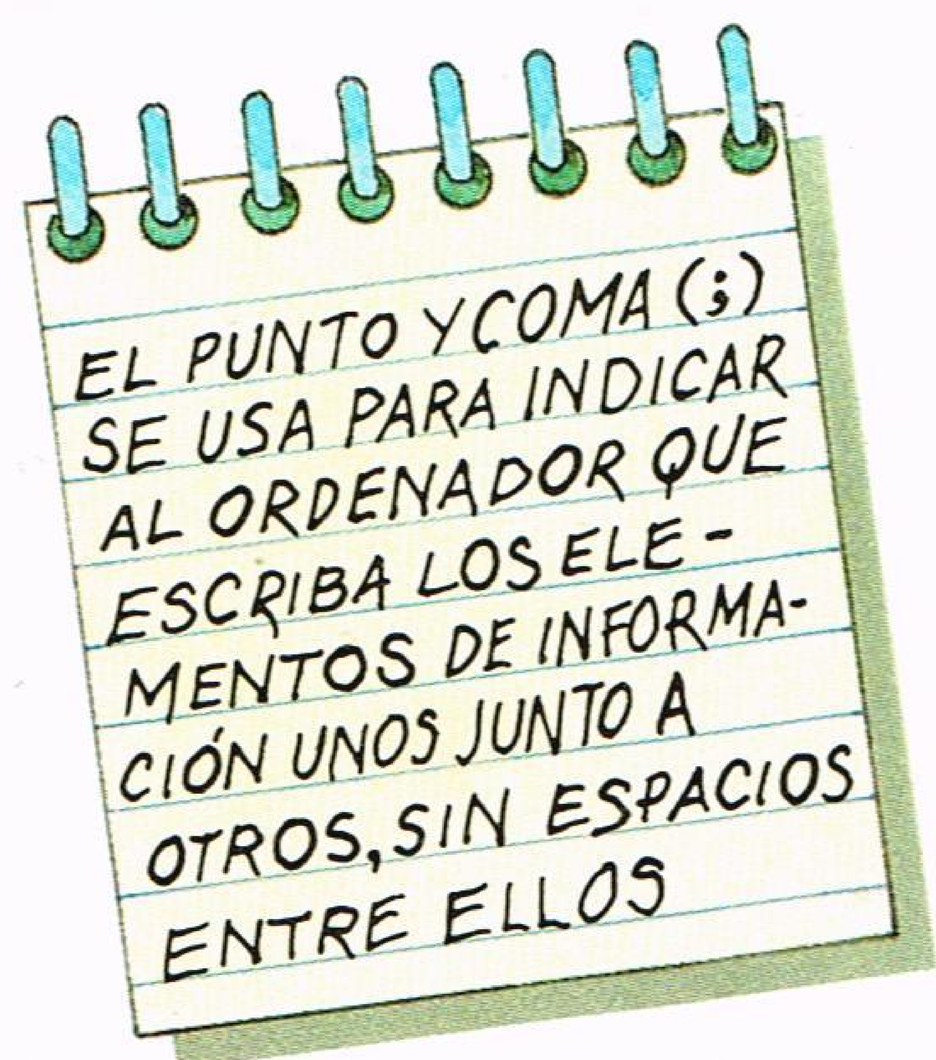
```
2000 REM TITULO
2010 PRINT"♥"
2020 FOR L=4 TO 10
2030 ON L-4 GOSUB 7500,7600,7700,7800,7900,8000
2040 NEXTL
2050 PRINT LEFT$(UD$,10)SPC(12)"▮AHORCADO"
2060 INPUT"▮▮▮▮▮▮▮▮▮▮▮▮▮           PULSA RETURN";A$
```

```
2070 PRINT"LETRAS ESCOGIDAS"
2080 PRINT"----------------"
2090 PRINT"LA PALABRA"
2100 PRINT"DEL ORDENADOR ES"
2110 PRINT"";
2120 FOR L =1 TO LO:PRINT"*";:NEXT L :PRINT
2130 PRINT"POR FAVOR, ESCOGE"
2140 PRINT"UNA LETRA"
2150 GA=0:PI=0:VI=0:PU=0
2160 RETURN
```

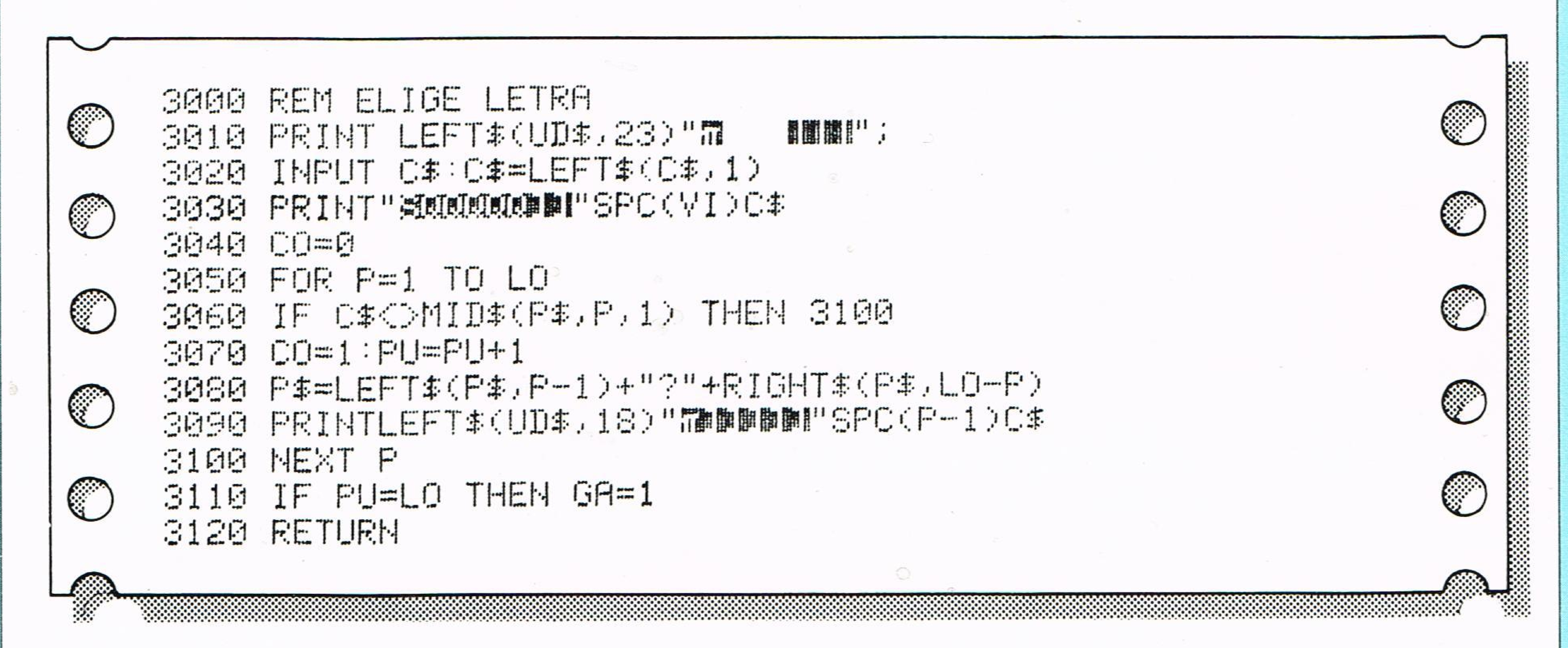
Las líneas de la 2070 a la 2160 completan la rutina de presentación. Primero, se borra la cabecera inicial del programa y se usa la instrucción **PRINT** para escribir el siguiente conjunto de instrucciones. Las letras elegidas durante el transcurso del juego aparecerán debajo de la cabecera, en lo alto de la pantalla. La palabra elegida por el ordenador se verá hacia la mitad de la pantalla. En la línea 2120 se escribe un asterisco por cada letra que contenga la palabra misteriosa. Esto se realiza utilizando otro bucle **FOR...NEXT**, cuya variable de control varía desde 1 hasta la longitud de la palabra obtenida por la subrutina *selecciona palabra*. Finalmente, el valor inicial de otras variables que se usan en el juego, como son **GA, PI, VI** y **PU** se ponen a cero. **GA** y **PI** se usan en el programa principal para llamar a la subrutina apropiada cuando se ha ganado o perdido el juego. **PU** se usa en el juego para comprobar si la palabra misteriosa ha sido correctamente adivinada, y **VI** para limitar el juego a diez intentos y construir la figura del ahorcado por etapas. Una vez más, puedes comprobar si has realizado correctamente esta parte del programa siguiendo las indicaciones del dibujo de abajo.



La siguiente subrutina realiza casi todo el trabajo del programa. Pide al jugador que elija una letra y comprueba si esa letra es una de las de la palabra misteriosa. Realiza los cambios apropiados en el valor de **PU** y comprueba si se ha ganado el juego. Como puedes suponer, ésta es la parte más difícil del programa.



La línea 3010 coloca el cursor en la línea 23 y cambia el color a amarillo. Tres espacios en blanco tapan la antigua entrada y el cursor se mueve hacia atrás tres espacios, hacia el comienzo de la línea. **INPUT C$** produce la aparición en la pantalla de un signo de interrogación y un cursor intermitente, para indicar al jugador que debe escribir una letra. La segunda parte de la línea 3020 asegura que **C$** sólo contenga un carácter. La línea 3030 escribe el intento en la fila quinta de la pantalla. La variable **VI** se usa para que el primer intento se escriba en el primer espacio, el segundo en el segundo, etc.



```
3000 REM ELIGE LETRA
3010 PRINT LEFT$(UD$,23)"■   ■■■■";
3020 INPUT C$:C$=LEFT$(C$,1)
3030 PRINT"■■■■■■■■■"SPC(VI)C$
3040 CO=0
3050 FOR P=1 TO LO
3060 IF C$<>MID$(P$,P,1) THEN 3100
3070 CO=1:PU=PU+1
3080 P$=LEFT$(P$,P-1)+"?"+RIGHT$(P$,LO-P)
3090 PRINTLEFT$(UD$,18)"■■■■■■"SPC(P-1)C$
3100 NEXT P
3110 IF PU=LO THEN GA=1
3120 RETURN
```

La línea 3040 pone a cero el valor de **CO** para que el programa vaya a la subrutina *ahorcado*, si la letra elegida no es correcta (vuelve a mirar la línea 60 del programa principal). Las líneas de la 3050 a la 3100 contienen un bucle **FOR...NEXT** que compara la letra elegida con cada una de las letras de la palabra misteriosa. Vamos a ver un ejemplo: supongamos que la palabra misteriosa es LIBRO y la letra que hemos introducido "A". **P** puede tomar valores entre uno y cinco, que es la longitud de la palabra elegida por el ordenador. **MID$** se usa para copiar parte de la palabra. Las instrucciones que van entre paréntesis nos indican que palabra y que parte de ésta se copia. Para cada uno de éstas, la línea 3060 comprobará que **C$** no es igual (<>) a **(P$,P,1)** así el programa podrá ir hasta el final del bucle y volver al programa principal. Si el intento fuese "R", el programa iría hasta **(P$,4,1).** Puesto que la letra elegida es la cuarta de las que forman la palabra LIBRO, el programa pasará a la línea 3070, donde se modifican los valores de **CO** y **PU**.

La línea 3080 coloca un signo de interrogación dentro de la palabra misteriosa, en el lugar de la letra adivinada, para que el jugador pierda una vida si hace el mismo intento otra vez. La línea 3090 escribe la letra adivinada en el lugar correcto dentro de la palabra misteriosa, y en la línea 3110 el ordenador compara el valor de **PU** con la longitud de la palabra. Si éstos son iguales, el valor de **GA** se hace igual a 1 y el ordenador vuelve al programa principal. Entonces, la línea 80 del programa principal llamará a la subrutina *ganar*.

```
LETRAS ESCOGIDAS
----------------

LA PALABRA
DEL ORDENADOR ES

B*FALO

POR FAVOR ESCOGE
UNA LETRA
```



PRUEBA DE LA SUBRUTINA *elige letra*, DEBE SER REALIZADA CON EL RESTO DEL PROGRAMA

1. AÑADE **REM** AL COMIENZO DE LAS LÍNEAS 60 Y 2030
2. ESCRIBE RUN PULSA RETURN

BORRA EL **REM** DE LAS LÍNEAS 60 Y 2030 ANTES DE CONTINUAR CON LA SIGUIENTE SECCIÓN

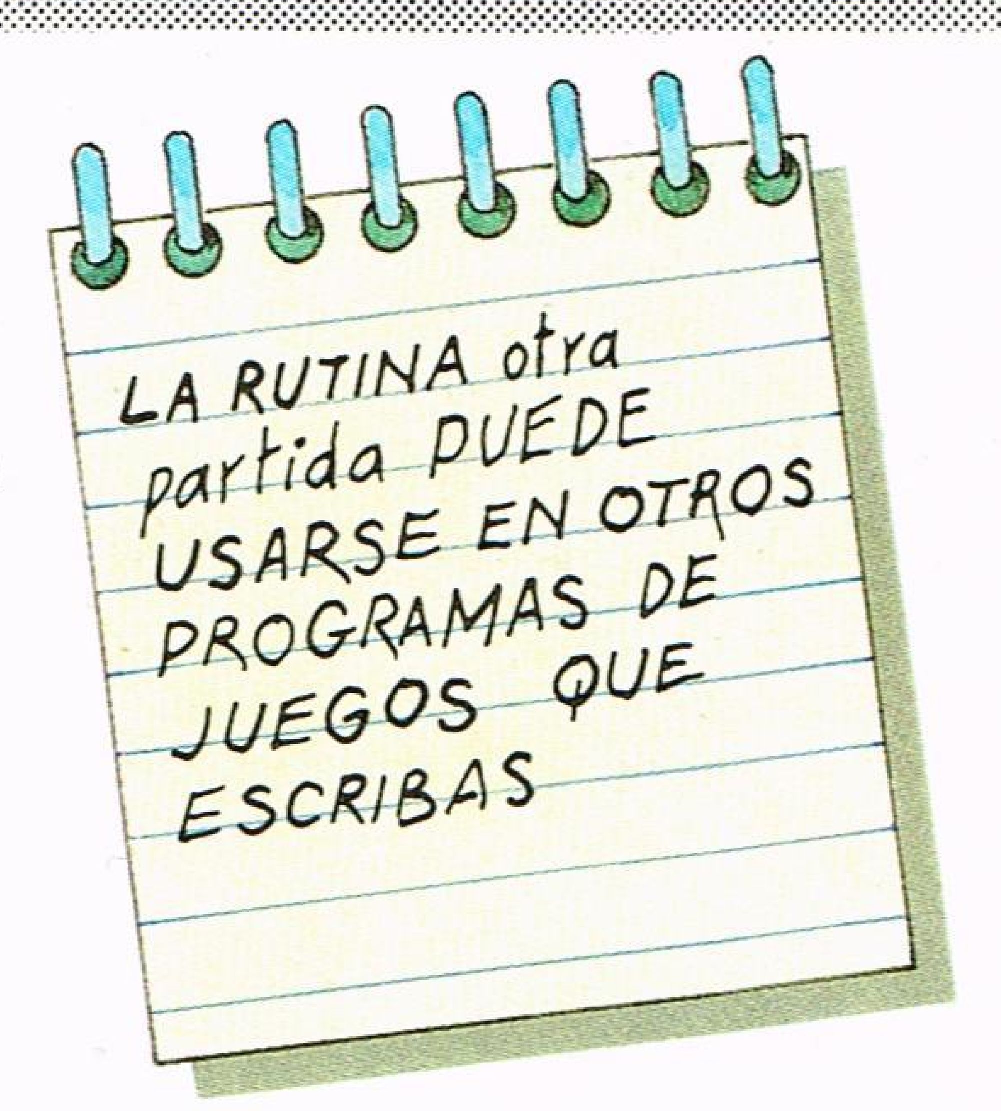
De la rutina *ahorcado* y de sus gráficos, se habla en la parte final del libro. Estos gráficos se usan en las subrutinas *ganar* y *perder*.

La rutina *ganar* es muy similar a la usada para conseguir la pantalla de presentación. La pantalla se borra y se dibujan los gráficos apropiados —el hombre— llamando a la rutina *ahorcado*. El mensaje de felicitación se escribe en la línea séptima usando **LEFT$(UD$,7)** y el color blanco. Recordarás que **UD$** se fijó en la línea 1005 para mover el cursor hacia abajo el número de líneas que se dan después de la coma; en este caso 7.

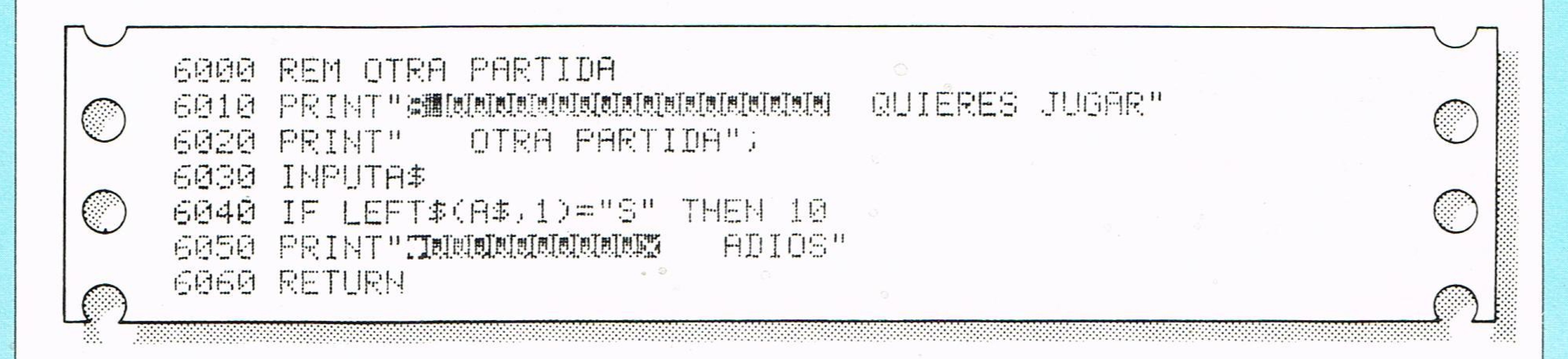
```
4000 REM GANAR
4010 PRINT"♥"
4020 FOR L = 4 TO 10
4030 ON L-4 GOSUB 7500,7600,7700,7800,7900,8000
4040 NEXT L
4050 PRINT LEFT$(UD$,7)"  ■BIEN HECHO ! TU EJECUCION"
4060 PRINT "  QUEDA APLAZADA"
4070 RETURN
```

```
5000 REM PERDER
5010 PRINT"♥"
5020 FOR L=1 TO 10
5030 ON L GOSUB 7100,7200,7300,7400,7500,7600,7700,
      7800,7900,8000
5040 NEXT
5050 PRINT "■■■■■■■■■  INUTIL !"
5060 PRINT"■    HAS PERDIDO !"
5070 PRINT"■■■■■LA PALABRA ERA■": PRINTSPC(5)D$
5080 RETURN
```

La rutina *perder* sigue casi exactamente la misma estructura que la *ganar*. En este caso, la totalidad de los gráficos del ahorcado aparecen en la pantalla. La palabra seleccionada originalmente por el ordenador, y almacenada como **D$** en la subrutina *selecciona palabra*, aparece en la pantalla mediante la línea 5070. Date cuenta de que cuando pides al ordenador que escriba una variable alfanumérica, no es necesario colocarla entre comillas.



La última subrutina pregunta al jugador si desea jugar otra vez. Es una rutina que podrás incluir en otros programas que realices. El **INPUT** de la línea 6030 indica que debemos introducir como entrada una letra que luego se comprobará (línea 6040) si es una **"S"**. Si ocurre esto, el programa volverá al comienzo del programa principal —línea 10— y el juego empezará otra vez. En caso contrario la pantalla se borrará y el ordenador dirá "adiós" en color rojo.



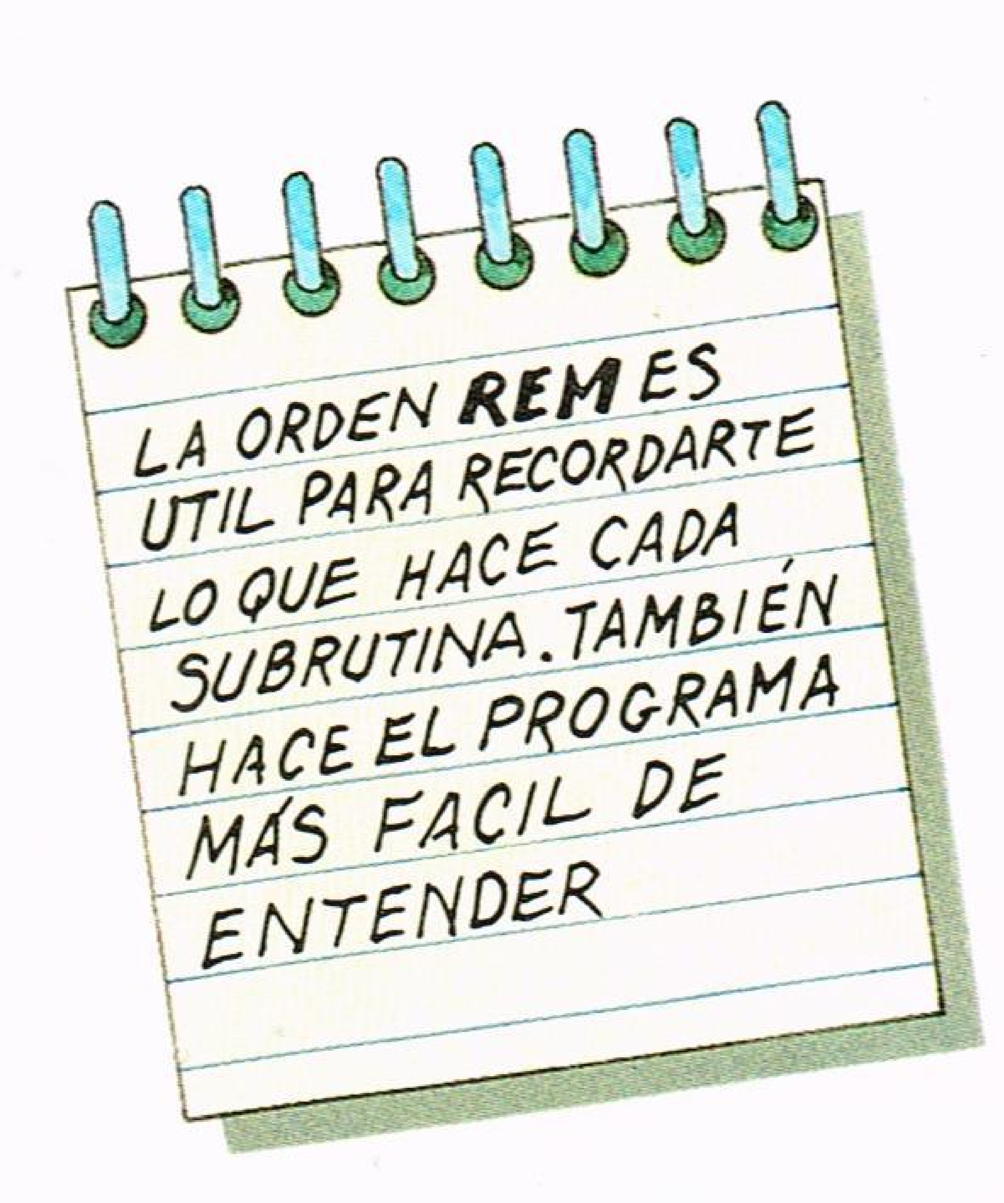
```
6000 REM OTRA PARTIDA
6010 PRINT"■■■■■■■■■■■■■■■■■■■■■  QUIERES JUGAR"
6020 PRINT"   OTRA PARTIDA";
6030 INPUTA$
6040 IF LEFT$(A$,1)="S" THEN 10
6050 PRINT"■■■■■■■■■■■■■■   ADIOS"
6060 RETURN
```

A excepción de los gráficos que contiene la subrutina *ahorcado*, se puede decir que el programa está ya completo. En este momento es una buena idea salvarlo en cinta. Conecta la grabadora como indique el manual, después escribe **SAVE "AHORCADO1"** y pulsa la tecla **RETURN.** El Commodore 64 te responderá escribiendo el siguiente mensaje **"PRESS PLAY RECORD ON TAPE".** Para saber si el programa ha sido grabado satisfactoriamente, rebovina la cinta y escribe **VERIFY "AHORCADO1"** y después pulsa **RETURN.** Si la grabación es buena, el ordenador responderá **"VERIFYING O.K.".** Si hubiera errores, salva (graba en cinta) el programa otra vez.

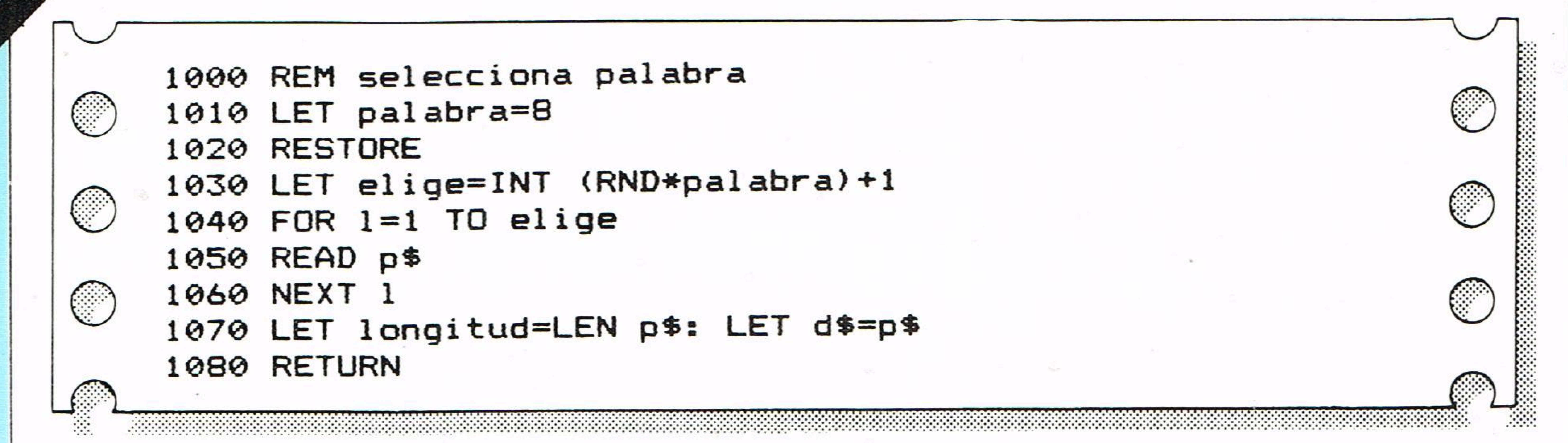
El programa principal del juego del ahorcado es similar al esquema que vimos en la introducción. Cada paso lógico se realiza en una subrutina. Cuando las subrutinas se colocan en orden, y se incluyen comprobaciones para ver si se ha ganado o perdido el juego, se obtiene el programa principal.

```
  10 CLS
  20 GO SUB 1000: REM selecciona palabra
  30 GO SUB 2000: REM titulo
  40 IF ganas OR pierdes=1 THEN  GO TO 80
  50 GO SUB 3000: REM elige letra
  60 IF correcto=0 THEN  GO SUB 7000: REM hombre ahorcado
  70 GO TO 40
  80 IF ganas=1 THEN  GO SUB 4000: REM ganar
  90 IF pierdes=1 THEN  GO SUB 5000: REM perder
 100 GO SUB 6000: REM otra partida ?
 110 STOP
```

Mira el programa principal con más detalle. Para empezar, va a una subrutina donde el ordenador elige una palabra aleatoria, de las almacenadas en el **DATA** del final del programa. Tras esto se escribe una pantalla de presentación. La línea 40 comprueba si el juego se ha ganado o se ha perdido (esto sólo ocurre después de que se haya hecho un intento). La línea 70 hace volver el programa al punto donde se introduce otra letra. La subrutina llamada en la línea 50 *(elige letra)* permite al jugador hacer otro intento. En la línea 60, el programa va a la rutina que contiene los gráficos del juego. El valor de *correcto* se pone a cero si se ha hecho un intento equivocado en la rutina *elige letra*. Por último, hay subrutinas que se utilizan cuando el juego se gana o se pierde, y otra que pregunta al jugador si quiere jugar otra partida. En once líneas tienes el programa completo.
Todo lo que queda es escribir las rutinas, y es aquí donde empieza el trabajo duro. Antes de hacer esto, añade en el **DATA** las palabras de abajo, de forma que puedas probar el programa a medida que lo vayas tecleando. El ordenador elegirá aleatoriamente una palabra de esta lista durante el transcurso del juego. Desde luego, no tienes por qué usar las palabras que damos aquí, pero en principio conviene que mantengas el número de palabras por debajo de ocho; podrás añadir más palabras cuando el programa esté completo.

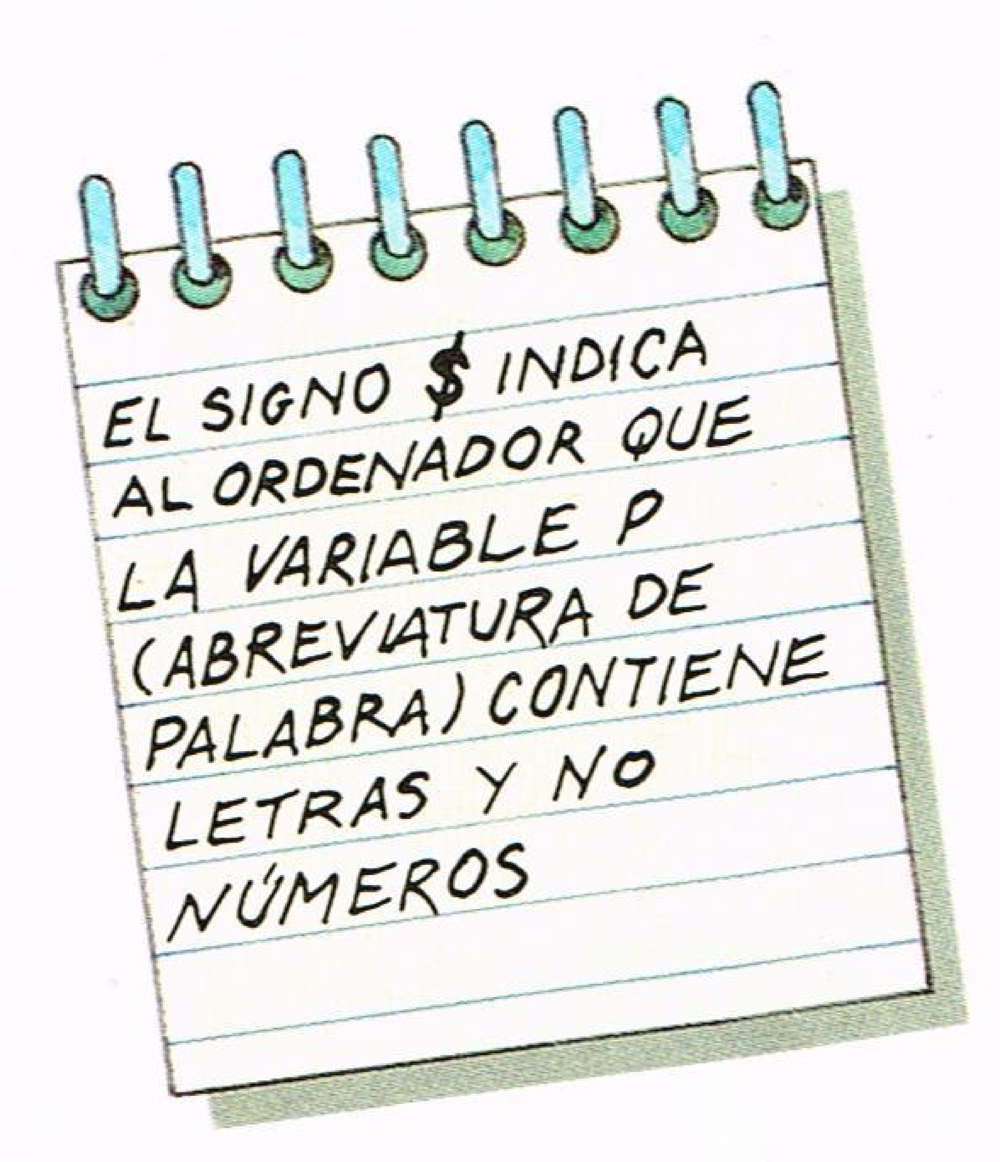


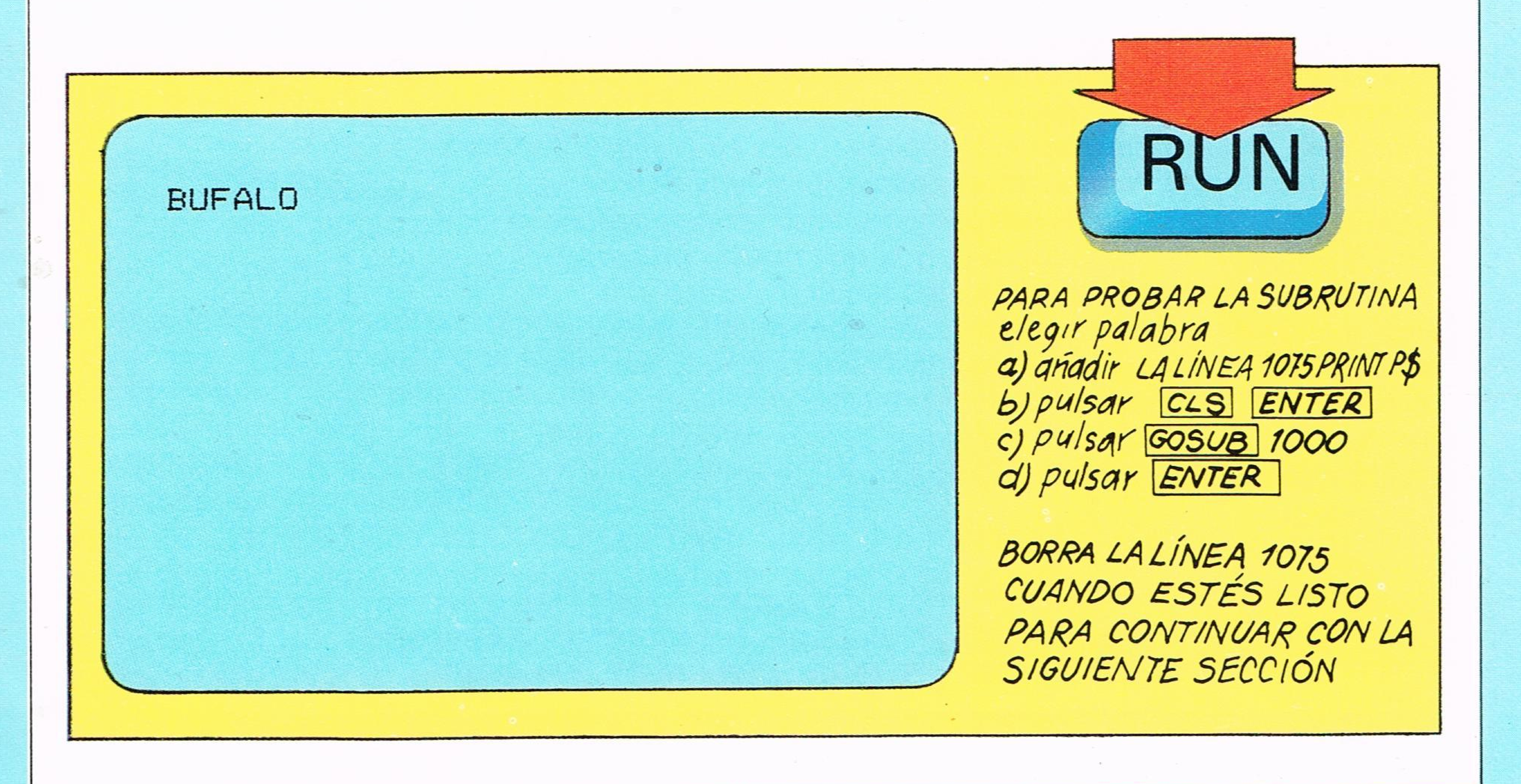

```
9000 DATA "MANZANA","ANGULO","ALFABETO","ANTIGUO"
9010 DATA "ALUBIAS","LIBRO","BUFALO","PLAYA"
```



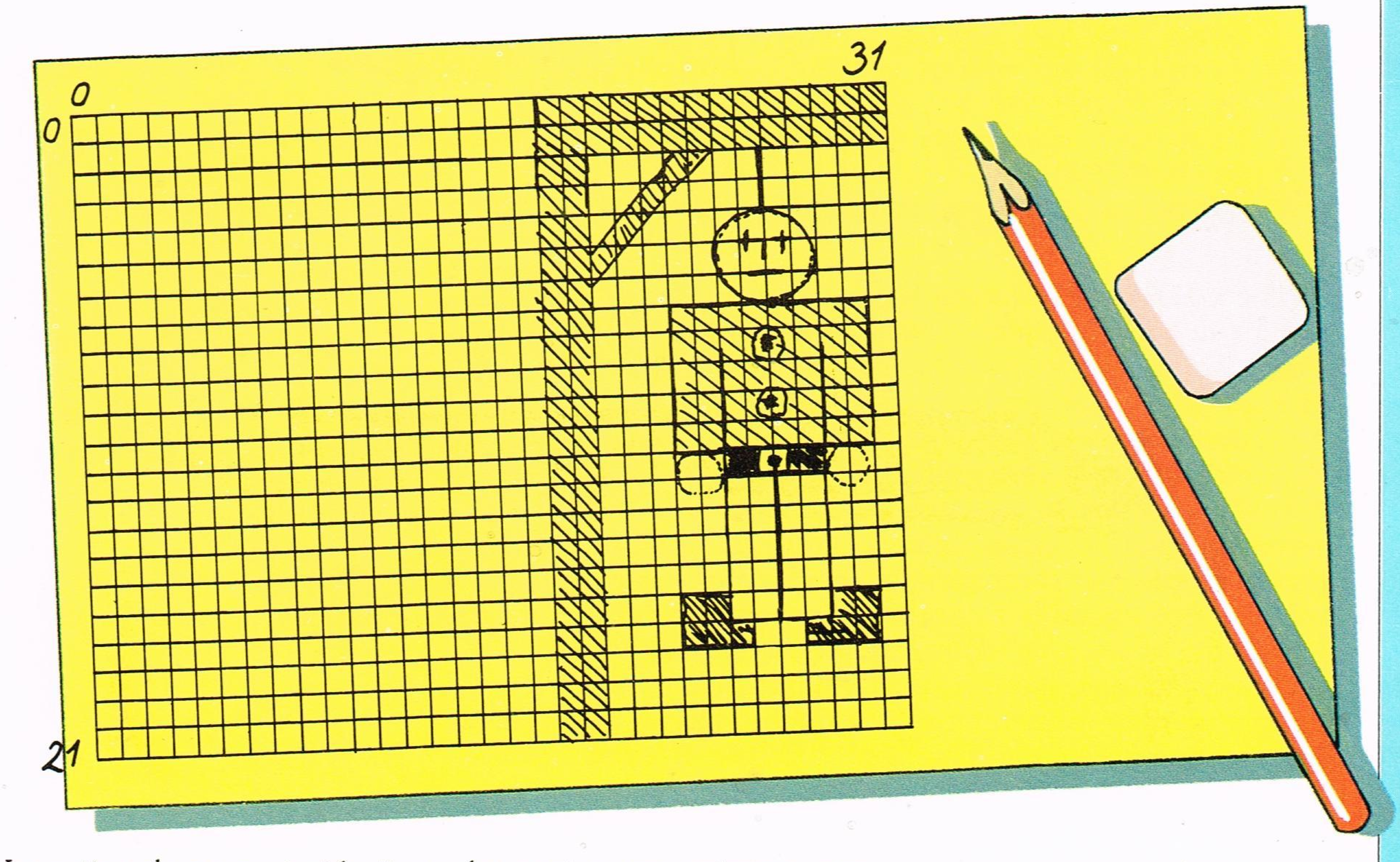
```
1000 REM selecciona palabra
1010 LET palabra=8
1020 RESTORE
1030 LET elige=INT (RND*palabra)+1
1040 FOR l=1 TO elige
1050 READ p$
1060 NEXT l
1070 LET longitud=LEN p$: LET d$=p$
1080 RETURN
```

La primera subrutina permite que el ordenador elija una palabra aleatoriamente de la lista almacenada en el **DATA**. En la línea 1010, la variable **palabra** toma el valor del número de palabras que hay en el **DATA**. En la línea 1020, la instrucción **RESTORE**, hace que el ordenador se coloque en el primer elemento del **DATA**, que será leído cada vez que se ejecute un **READ**. La siguiente línea, la 1030, genera un número aleatorio entre uno y el número de palabras del **DATA**. El Spectrum, en principio, genera números aleatorios entre 0 y 1, pero luego el número generado se multiplica por el número de palabras del **DATA**, y se convierte en un número entero por medio de **INT**. Se suma 1 para evitar que el número pueda ser cero. El bucle **FOR...NEXT,** de las líneas 1040 a 1060, hace que el ordenador lea el **DATA** hasta el punto indicado por el número aleatorio. Por ejemplo, si el número fuese 4, el ordenador leería hasta la cuarta palabra. Esta es la palabra que habrá de ser adivinada, y se almacena en **P$.** La línea 1070 calcula su longitud y almacena la palabra en **d$** (**d$** se usa al final de un juego perdido, cuando el ordenador escribe la palabra misteriosa).





La subrutina siguiente escribe la pantalla de presentación; para ello usará los gráficos de la subrutina *ahorcado*. Estos gráficos aparecerán en el lado derecho de la pantalla; habremos de tenerlo en cuenta para decidir dónde colocar el texto en la pantalla.



La rutina de presentación tiene dos partes, una primera, que se da abajo, y que nos proporciona la pantalla inicial, y una segunda que nos deja la pantalla lista para el comienzo del juego. La línea 2010 borra la pantalla de todo lo que pudiera contener. La línea 2020 manda el programa a la sección gráfica apropiada dentro de la subrutina *ahorcado*. Todo esto quedará más claro cuando lleguemos a aquella parte del programa. En resumen, la imagen del ahorcado se construye a lo largo de diez etapas, una por cada intento fallido o «vida» perdida. Para la pantalla de presentación sólo usaremos los gráficos creados por las etapas quinta a décima. En la línea 2030, la instrucción **PAPER** da el color del fondo de la pantalla e **INK** da el color del texto. La instrucción **PRINT AT** coloca la palabra "AHORCADO" en la quinta línea de la pantalla, a 14 columnas del margen izquierdo. Entonces se le indica al jugador que pulse **ENTER** para comenzar el juego.

```
2000 REM titulo
2010 CLS
2020 FOR l=5 TO 10: GO SUB 7000+l*100: NEXT l
2030 PAPER 7: INK 0: PRINT AT 4,2;"H O M B R E
     A H O R C A D O"
2040 PRINT AT 21,5;"pulsa ENTER para empezar": INPUT a$
```

```
2045 CLS
2050 PRINT AT 2,2;"Letras escogidas"
2060 PRINT AT 3,2;"----------------"
2070 PRINT AT 12,2;"La palabra"
2080 PRINT AT 14,2;"del ordenador es"
2090 INK 2: PRINT AT 17,2;
2100 FOR l=1 TO longitud: PRINT "*";: NEXT l
2110 INK 0: PRINT AT 20,2;"Por favor, escoge"
2120 PRINT AT 21,6;"una letra"
2130 LET ganas=0: LET pierdes=0: LET vida=0: LET puntos=0
2140 RETURN
```

Las líneas 2050 a 2140 completan la rutina de presentación. Primero se borra la pantalla inicial y la instrucción **PRINT AT** se usa para escribir el siguiente conjunto de instrucciones. Las letras elegidas en el transcurso del juego aparecerán debajo de la cabecera, en lo alto de la pantalla. Si se gana, la palabra seleccionada aparecerá en mitad de la pantalla. En la línea 2100 se imprimirá un asterisco por cada letra de la palabra misteriosa. Esto se realiza utilizando otro bucle **FOR...NEXT,** con la variable que controla el bucle oscilando entre 1 y la longitud obtenida en la subrutina *selecciona palabra.* Finalmente, el valor de alguna de las otras variables utilizadas en el juego **(ganas, pierdes, vida** y **puntos)** se pone a cero. **Ganas** y **pierdes** se usan en el programa principal para llamar a la subrutina apropiada, si se gana o pierde el juego. **Puntos** se usa durante el juego para saber si la palabra misteriosa ha sido correctamente adivinada, y **vida** se usa para limitar el juego a diez intentos y para construir el gráfico del ahorcado por etapas. Una vez más, puedes comprobar si has realizado correctamente esta parte del programa siguiendo las instrucciones del dibujo de abajo.

PRUEBA DE LA SUBRUTINA *título* a) AÑADE LA LÍNEA: **2095 LET longitud= 8** b) AÑADE **REM** AL COMIENZO DE LA LÍNEA 2020 c) PULSA [CLS] [ENTER] d) PULSA [GOSUB] 2000 e) PULSA [ENTER] BORRA LO AÑADIDO CUANDO ESTÉS LISTO PARA CONTINUAR CON LA SIGUIENTE PARTE DEL PROGRAMA




```
EL   AHORCADO



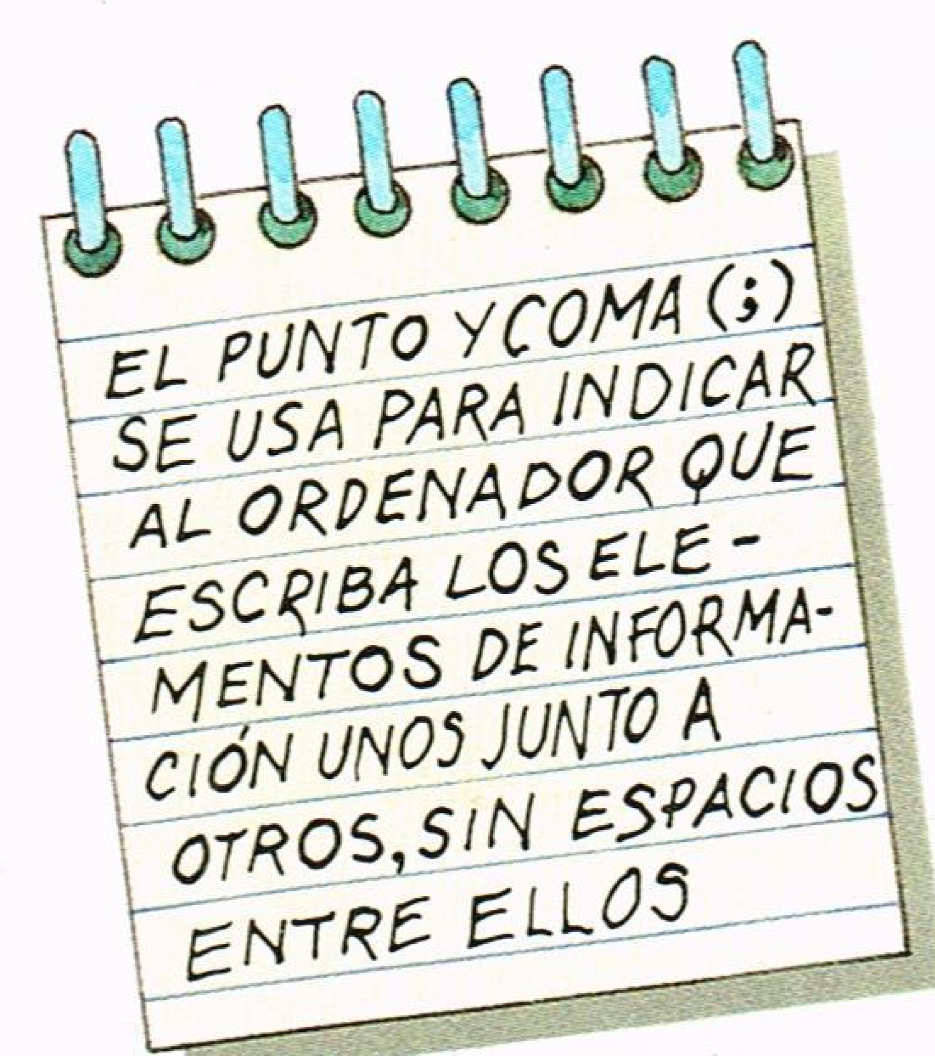
Pulsa ENTER para empezar
```

```
Letras escogidas
----------------

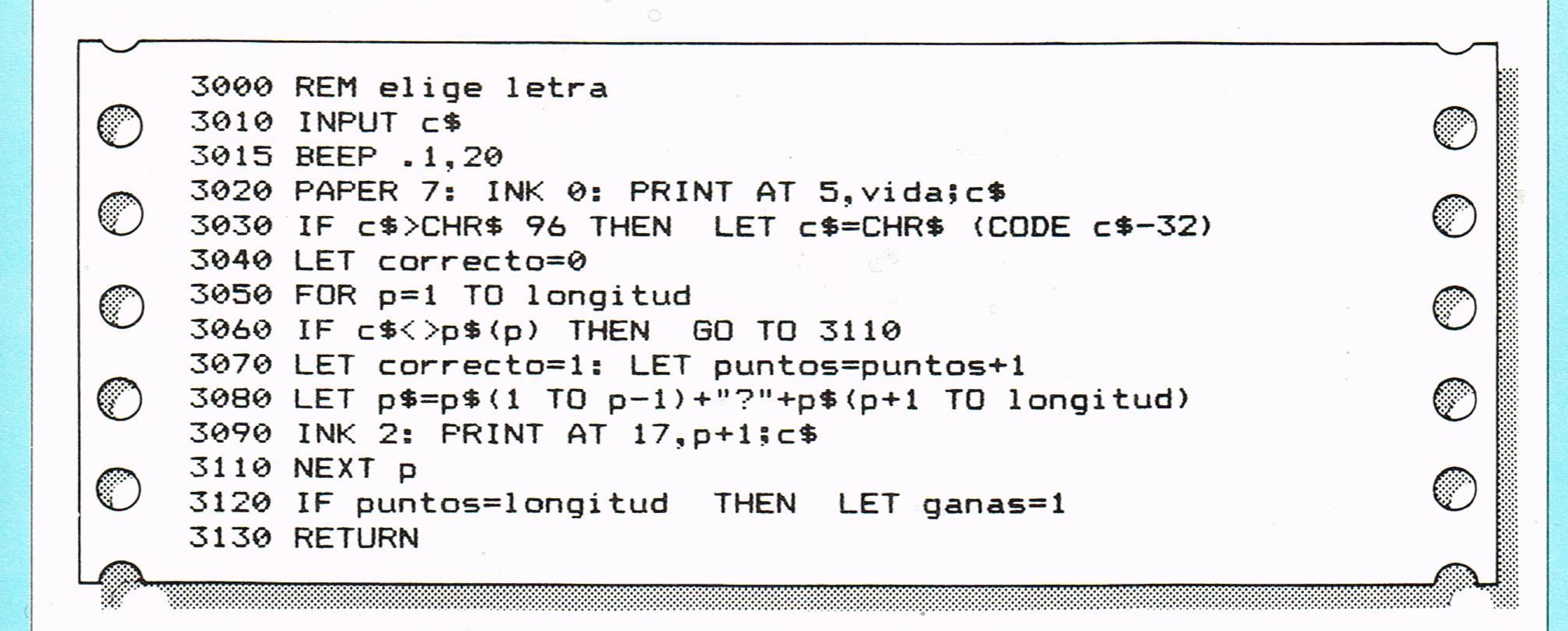

La palabra
del ordenador es
********

Por favor, escoge
una letra
```

La siguiente subrutina realiza casi todo el trabajo del programa. Para empezar pide al jugador que elija una letra y comprueba si ésta forma parte de la palabra misteriosa; además realiza los cambios adecuados en el valor de **puntos** y comprueba si se ha ganado el juego.



La línea 3010 pide la introducción de la letra elegida, mediante un **INPUT.** Esta instrucción hace que aparezca un cursor intermitente en la pantalla, para indicar al usuario que está esperando la introducción de un dato. La línea 3020 escribe la letra leída en la quinta línea de la pantalla; la variable **vida** se utiliza para que la primera letra se escriba en el primer espacio, la segunda en el segundo, etc. La línea 3030 asegura que el ordenador acepte tanto minúsculas como mayúsculas. Todas las teclas de caracteres tienen códigos numéricos **CHR$** (el Spectrum, por ejemplo, utiliza el código ASCII), por ejemplo **CHR$ 84** es el código de la "T" y **CHR$ 116** el de la "t". Esta línea, por tanto, convierte las letras minúsculas en mayúsculas, restándole a su código 32.



```
3000 REM elige letra
3010 INPUT c$
3015 BEEP .1,20
3020 PAPER 7: INK 0: PRINT AT 5,vida;c$
3030 IF c$>CHR$ 96 THEN  LET c$=CHR$ (CODE c$-32)
3040 LET correcto=0
3050 FOR p=1 TO longitud
3060 IF c$<>p$(p) THEN  GO TO 3110
3070 LET correcto=1: LET puntos=puntos+1
3080 LET p$=p$(1 TO p-1)+"?"+p$(p+1 TO longitud)
3090 INK 2: PRINT AT 17,p+1;c$
3110 NEXT p
3120 IF puntos=longitud  THEN  LET ganas=1
3130 RETURN
```

La línea 3040 pone el valor de **correcto** a cero, para que el programa vaya a la subrutina *ahorcado* si no se ha elegido una letra válida. (Vuelve atrás y mira la línea 60 del programa principal.) Las líneas de la 3050 a la 3110 contienen un bucle **FOR...NEXT** que compara la letra leída con cada una de las letras de la palabra misteriosa. Por ejemplo, supongamos que la palabra misteriosa es LIBRO y que la letra leída es "A". Entonces la variable **p** podrá tomar valores desde uno hasta cinco; **p$(1)** será L, **p$(2)** será I, **p$(3)** B, **p$(4)** R, y **p$(5)** será O. Para cada uno de éstos, la línea 3060 comprobará que **c$** no es igual (< >) a **p$(p),** hasta que se acabe el bucle y se vuelva al programa principal. Si la letra fuese O el bucle se ejecutaría hasta **p$(5)** e iría a la línea 3070, donde se cambian los valores de **correcto** y de **puntos.**

La línea 3080 coloca una interrogación en el lugar de la palabra misteriosa que ocupa la letra adivinada, para que, si el jugador hace el mismo intento otra vez, pierda una vida. La línea 3090 escribe la letra adivinada en el lugar que ocupa dentro de la palabra misteriosa, y en la línea 3120 el ordenador compara el valor que tiene **puntos** con la longitud de la palabra. Si estos valores son iguales, se da el valor 1 a la variable **ganas** y el ordenador vuelve al programa principal. Entonces, la línea 80 del programa principal llamará a la subrutina *ganar*.

```
L e t r a s   e s c o g i d a s
--------------------------------



L a   p a l a b r a
d e l   o r d e n a d o r   e s

A * * * * *

P o r   f a v o r , e s c o g e
u n a   l e t r a
```

RUN

PRUEBA DE LA SUBRUTINA *elige letra* : PARA ELLO ES NECESARIO EJECUTARLA CON EL RESTO DEL PROGRAMA
a) AÑADE **REM** AL PRINCIPIO DE LAS LÍNEAS 60 Y 2020
b) PULSA CLS ENTER
c) PULSA RUN ENTER
BORRA LOS **REM** DE LAS LÍNEAS 60 Y 2020 ANTES DE SEGUIR

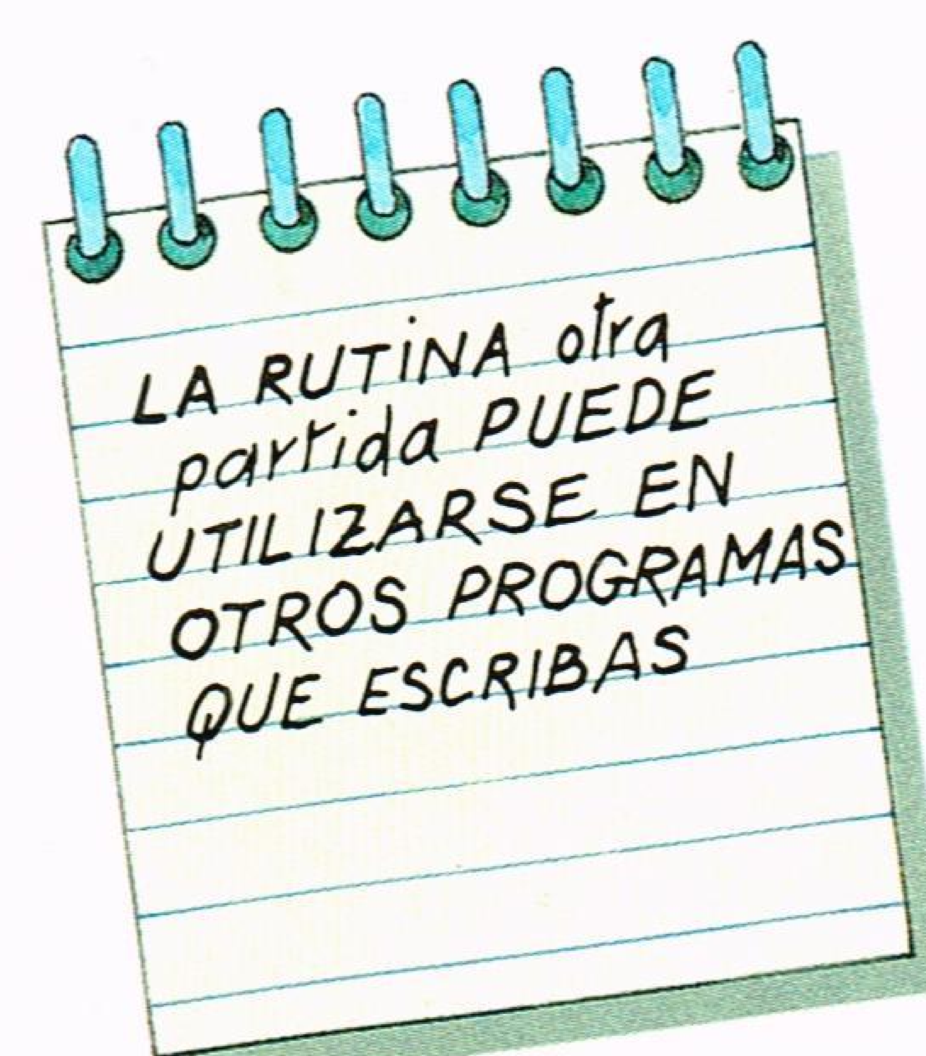
De la rutina *ahorcado* y los gráficos que contiene, se trata en la parte final del libro. Estos gráficos se usan en las subrutinas *ganar* y *perder*.

La rutina *ganar* es muy parecida a la usada para conseguir la pantalla de presentación. La pantalla se borra, y los gráficos apropiados —el hombre— se dibujan llamando a la rutina *ahorcado*. Las instrucciones **INK** y **PAPER** se usan para dar color al texto y al fondo. Podrías añadir una instrucción **FLASH** para conseguir que el texto sea intermitente; esto se hace colocando la instrucción **FLASH 1**, seguida de dos puntos, al principio de la línea 4030. Para terminar con la intermitencia, coloca la instrucción **FLASH 0**, al final de la línea 4040.

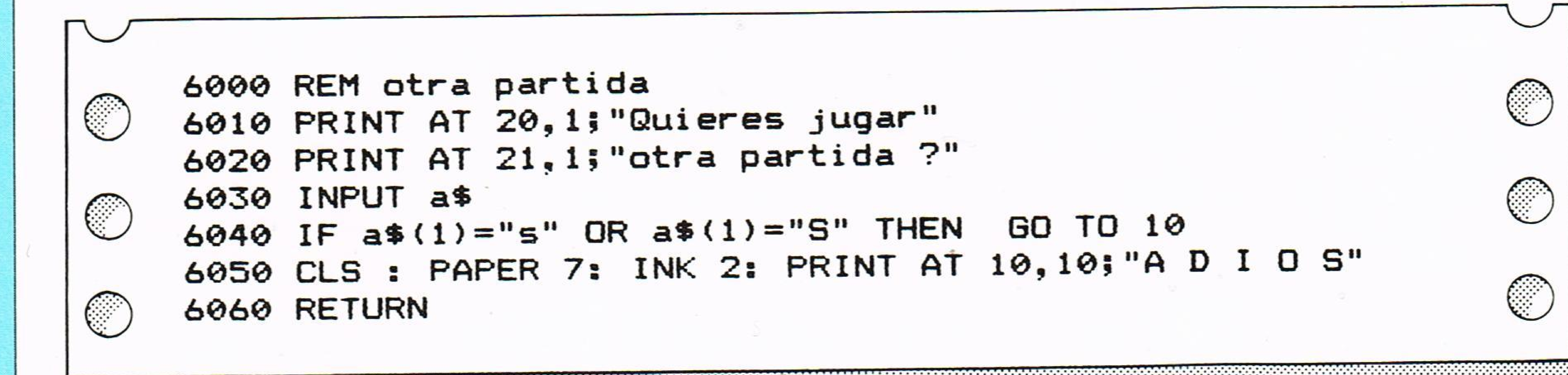
```
4000 REM ganar
4010 CLS
4020 FOR l=5 TO 10: GO SUB 7000+l*100: NEXT l
4030 INK 0: PAPER 7: PRINT AT 1,4;"Bien hecho ! Tu ejecucion"
4040 PRINT AT 2,4;"queda aplazada"
4045 PLOT 222,110: DRAW -3,3
4046 PLOT 232,110: DRAW 3,3
4047 FOR s=0 TO 40: BEEP .05,s: NEXT s
4050 RETURN
```

```
5000 REM perder
5010 PAPER 7: INK 0: CLS
5020 FOR l=1 TO 10: GO SUB 7000+l*100: NEXT l
5030 INK 2: PAPER 7: PRINT AT 7,2;"Inutil !"
5040 PRINT AT 12,2;"Has perdido !"
5050 PRINT AT 14,2;"La palabra era ";: PRINT AT 16,5;d$
5060 RETURN
```

La rutina *perder* tiene casi exactamente la misma estructura que la *ganar*. En caso de perder el juego, la totalidad de los gráficos del ahorcado aparecen en la pantalla. La palabra seleccionada originalmente por el ordenador, y que se almacenó en la variable **d$** cuando se llamó a la rutina *elegir palabra*, aparece en la pantalla por medio de la línea 5050. Date cuenta de que cuando pides al ordenador que imprima una variable alfanumérica no es necesario colocarla entre comillas.

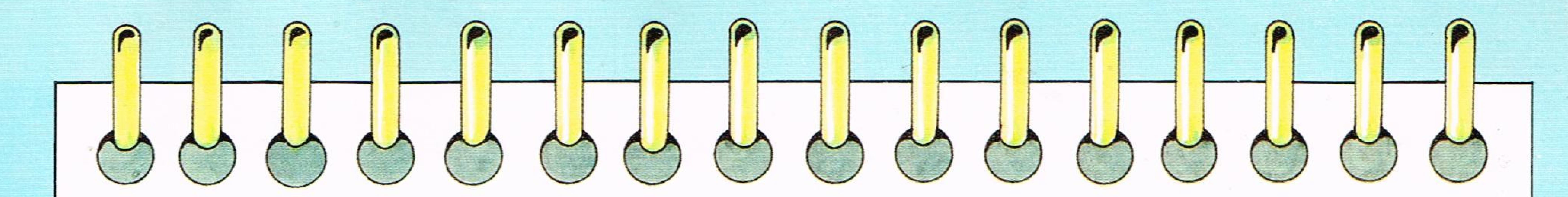


La última subrutina pregunta al jugador si quiere jugar otra partida. Esta rutina la puedes utilizar en cualquier otro programa que tú escribas. En la línea 6030 se lee una letra con un **INPUT** y en la 6040 se comprueba si es una "S", bien mayúscula o minúscula. En ese caso, se salta al principio del programa principal —línea 10— y el juego comienza de nuevo. En cualquier otro caso, la pantalla se borra y el ordenador dice "adiós".



```
6000 REM otra partida
6010 PRINT AT 20,1;"Quieres jugar"
6020 PRINT AT 21,1;"otra partida ?"
6030 INPUT a$
6040 IF a$(1)="s" OR a$(1)="S" THEN  GO TO 10
6050 CLS : PAPER 7: INK 2: PRINT AT 10,10;"A D I O S"
6060 RETURN
```

El programa está ya terminado, a excepción de los gráficos contenidos en la rutina *ahorcado*. En este momento, sería una buena idea salvarlo en cinta, conectando la grabadora como indica el manual del ordenador. Teclea **SAVE "AHORCA1"** y pulsa **ENTER**. Aparecerá el mensaje **"Start tape then press any key"**. Para comprobar que el programa se ha grabado bien, rebobina la cinta y teclea **VERIFY "AHORCA1"** (**VERIFY** está en la tecla R), pulsa "play" en la grabadora y si la grabación fue correcta saldrá el mensaje **"O.K."**. En caso contrario, saldrá un mensaje de error. En este último caso debes volver a salvar (grabar en cinta) el programa.



Mejora tus programas
Una vez que hayas escrito el programa, probablemente te sabrás de memoria las palabras de la lista **DATA**. Confecciona con un amigo una lista nueva de palabras o, mejor todavía, id poniendo una palabra cada uno sin que el otro las pueda ver. Toma nota del número de palabras que habéis introducido y modifica, según éste, el valor de la variable de la línea 1010.

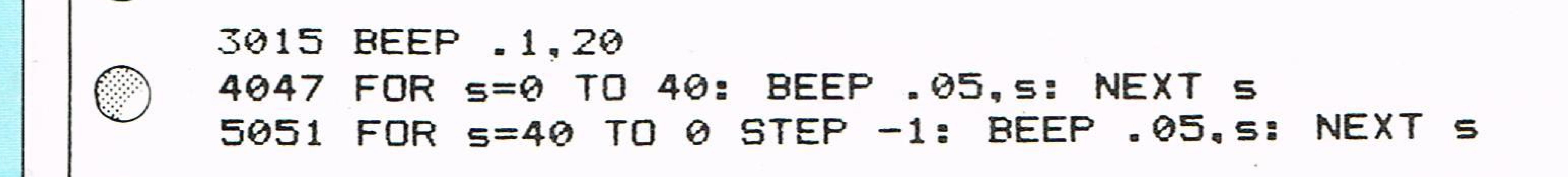


Añade sonido

Commodore
El Commodore 64 está dotado de un poderoso sintetizador musical, pero sacarle todo su partido se encuentra fuera del alcance de este libro. El apartado de abajo muestra cómo se utiliza la variable del bucle **FOR...NEXT** para obtener el valor que va a ser introducido **(POKE)** en el registro de frecuencia **HI,** con objeto de hacer variar el tono **(POKE 54273,I).** Las notas decrecen cuando el juego se pierde, y crecen cuando se gana. Consulta tu manual para más detalles.

```
4063 FOR I=0 TO 24:POKE 54272,0:NEXT I
4067 POKE 54277,15:POKE 54278,240:POKE 54296,15
4068 POKE 54276,33:FOR I=0 TO 255:POKE 54276,I: NEXT
4069 POKE 54276,32
```

Spectrum
La instrucción de sonido del Spectrum es **BEEP**, seguido de dos números; el primero controla la duración del sonido y el segundo el tono. Para conseguir una sucesión de sonidos, puedes usar una serie de instrucciones **BEEP** separadas por dos puntos (:). En el apartado de abajo se usa un bucle **FOR...NEXT** para hacer variar el valor del tono de la instrucción **BEEP**, con objeto de conseguir sonidos distintos. Consulta tu manual para más detalles sobre cómo producir sonidos.

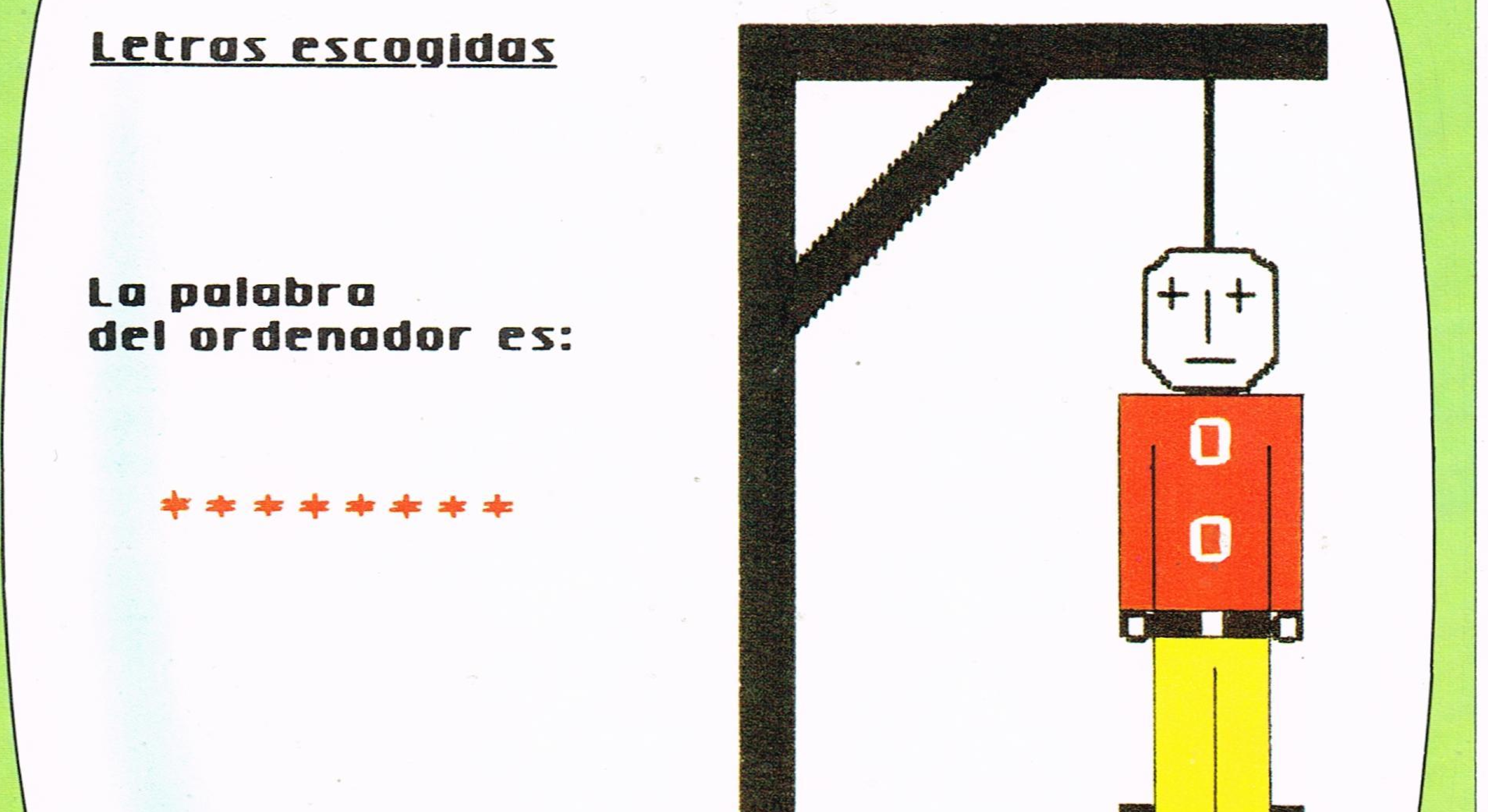


```
3015 BEEP .1,20
4047 FOR s=0 TO 40: BEEP .05,s: NEXT s
5051 FOR s=40 TO 0 STEP -1: BEEP .05,s: NEXT s
```

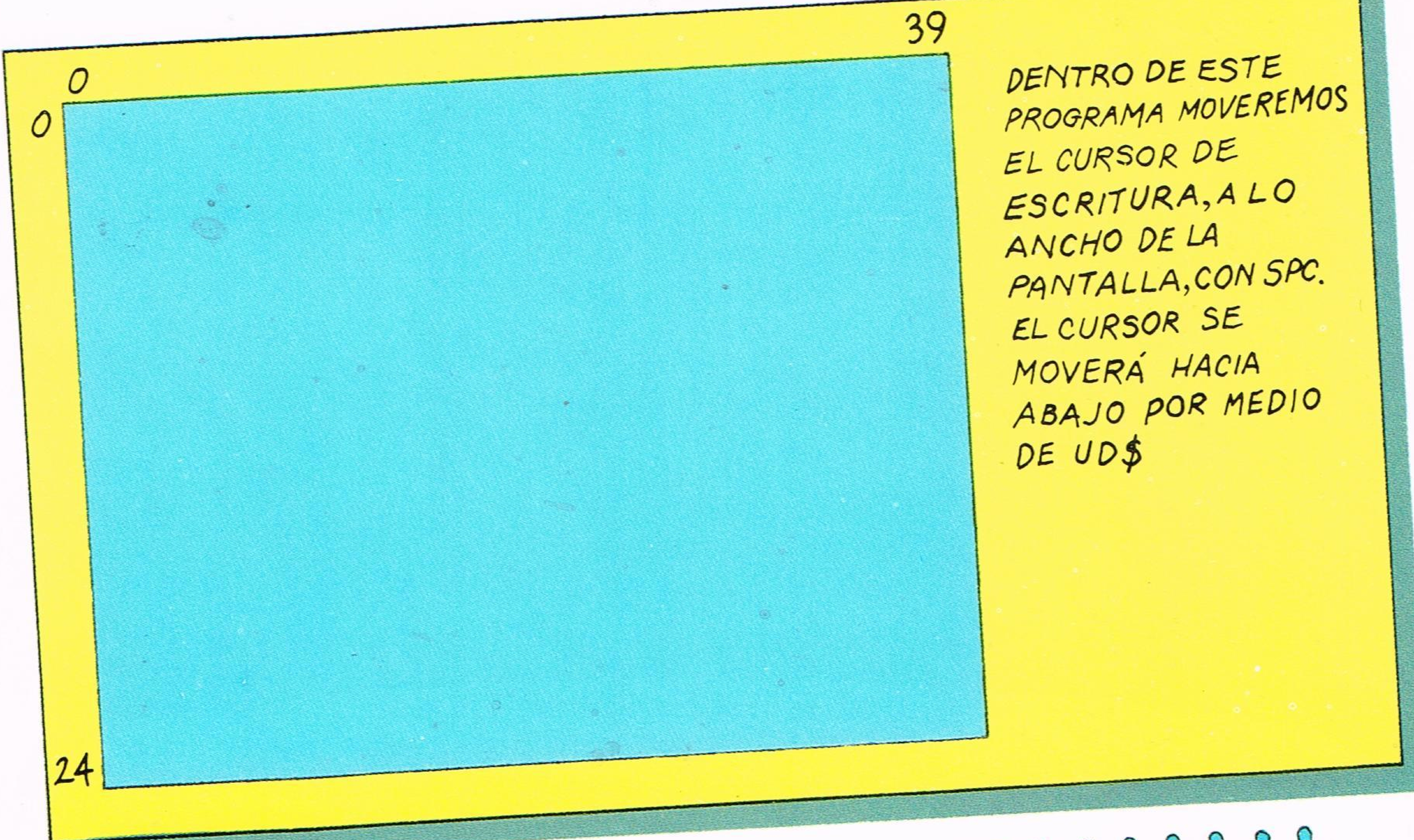


LOS GRÁFICOS

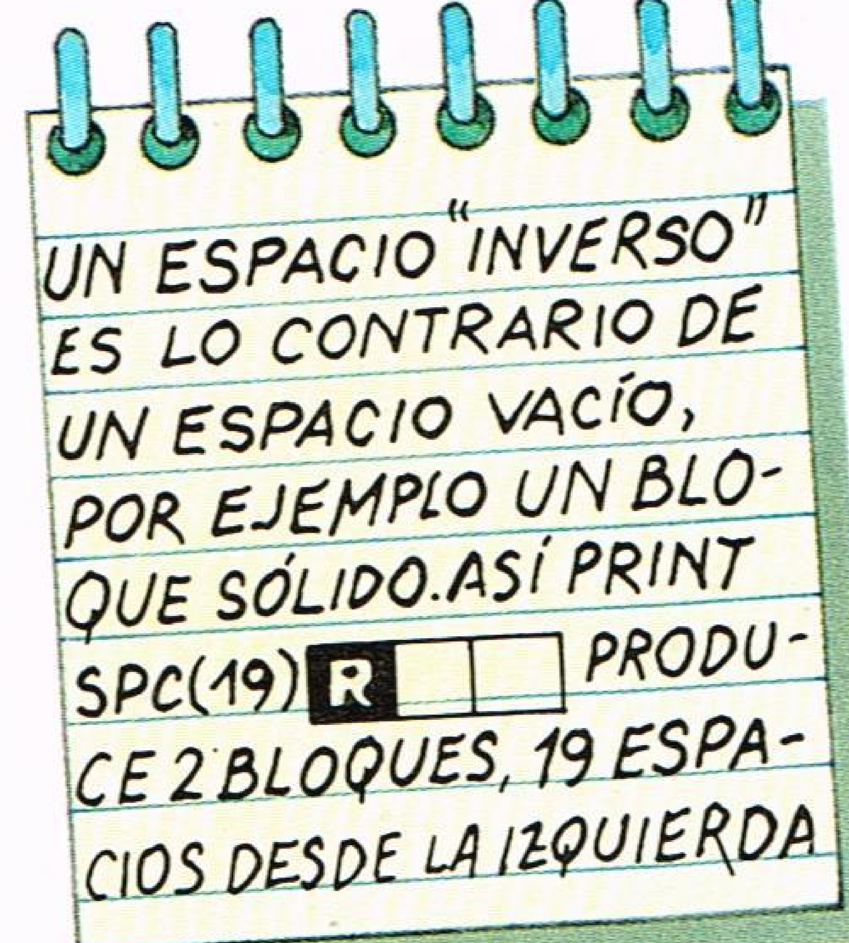
Esta sección del libro muestra los gráficos que se usan en el juego del ahorcado. El dibujo se construye en diez etapas, una por cada vida perdida en el curso del juego. Las instrucciones gráficas usadas te enseñan cómo se pueden dibujar líneas en la pantalla y cómo colorear zonas de la pantalla con los colores de que dispone el ordenador.



La pantalla de textos del Commodore 64 se divide en 40 líneas y 25 columnas verticales. Para mover el cursor de escritura hacia abajo, en la pantalla, usaremos **UD$** (o parte de ella). Para moverlo horizontalmente usaremos **SPC.**

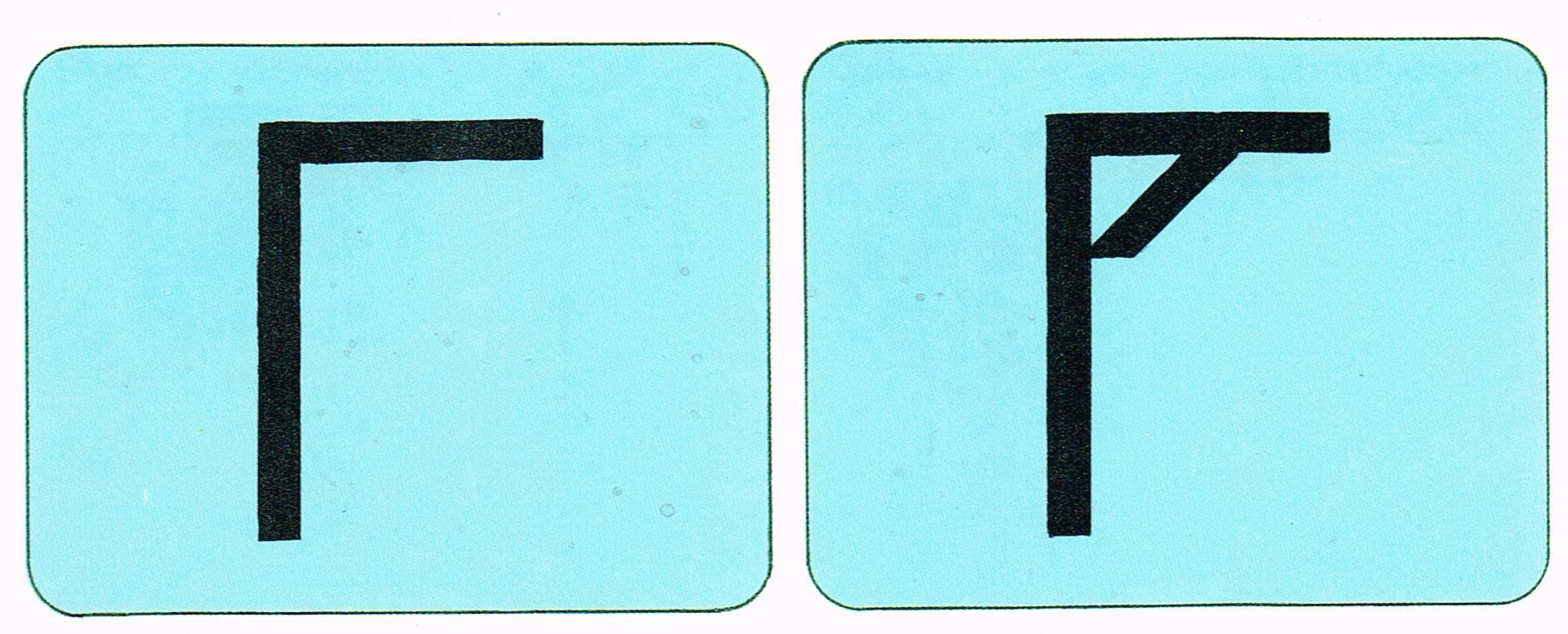


La subrutina *ahorcado* se divide en diez etapas, que irán apareciendo en pantalla cuando se realice un intento fallido y se pierda una vida. En la línea 7010, el valor de **VI** se incrementa en uno cada vez que se llama a la rutina, y se comprueba si este valor sobrepasa el de diez, en cuyo caso el juego se ha perdido. La línea 7020 dirige el programa a la sección apropiada de los gráficos usando el valor de **VI**. La primera sección de los gráficos levanta el poste de la horca, usando un sencillo gráfico de bloques obtenido por modo **INVERSO** y dos espacios. El inverso de un espacio es un bloque y puede usarse con cualquier color.

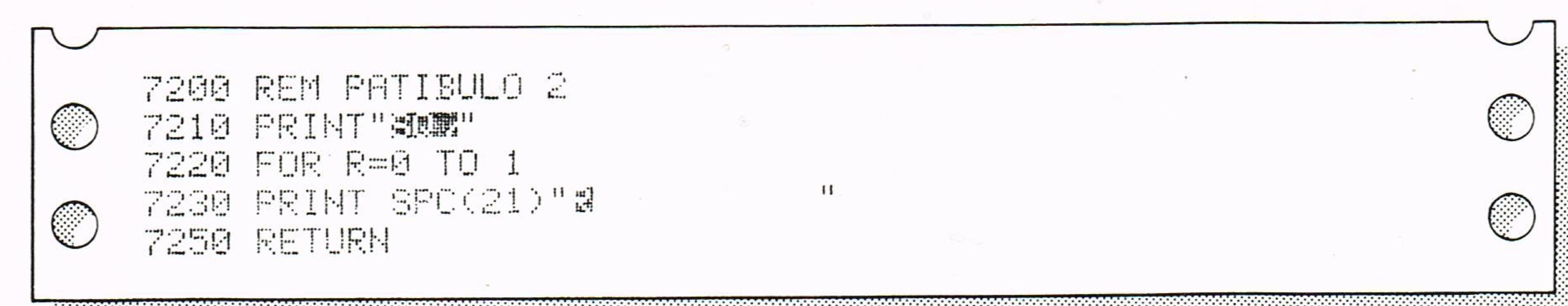



```
7000 REM HOMBRE AHORCADO
7010 VI=VI+1:IF VI=10 THEN PI=1
7020 ON VI GOTO 7100,7200,7300,7400,7500,7600,7700,
     7800,7900,8000
7100 REM PATIBULO 1
7110 PRINT"[illegible]"
7120 FOR R=0 TO 21
7130 PRINT SPC(19)"[illegible]  "
7140 NEXT
7150 RETURN
```

Hay tres símbolos en la línea 7110: el primero coloca el cursor, el segundo lo mueve hacia abajo y el último selecciona el color marrón. El cursor *home* se representa por el inverso "S". El movimiento del cursor hacia abajo se representa por el inverso "Q" y el marrón se obtiene con 2. El bucle **FOR...NEXT** selecciona las líneas de la 0 a la 21, y dibuja dos bloques, comenzando en la columna 19. Para probar esta primera sección de gráficos escribe **GOSUB 7100.**



El segundo apartado de los gráficos añade al dibujo la barra horizontal de la horca. Usa exactamente el mismo método que el empleado para levantar el poste. Las líneas de la 7220 a la 7240 construyen un bloque de once espacios de anchura, en dos filas.



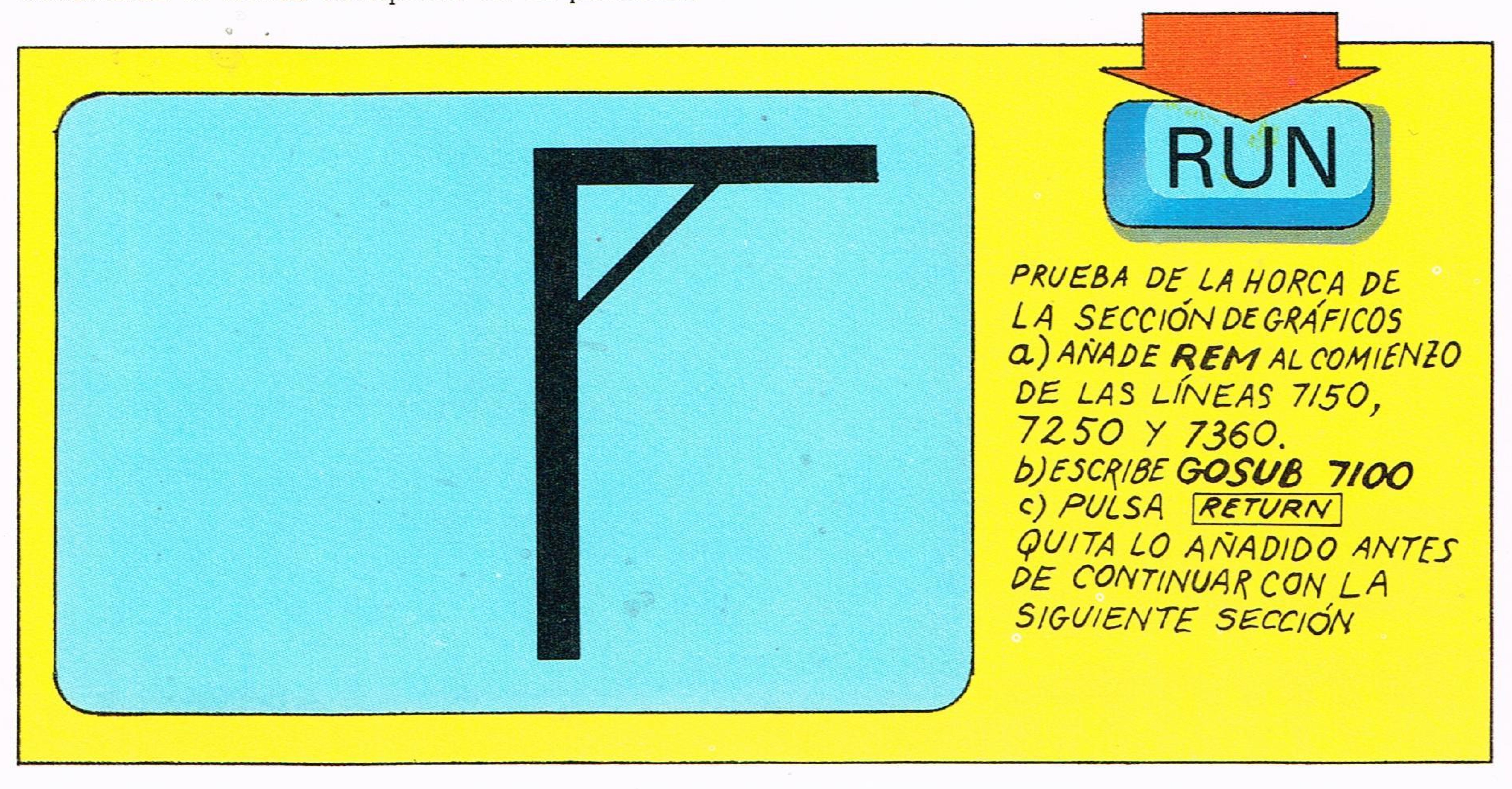
```
7200 REM PATIBULO 2
7210 PRINT"▒▒▒▒"
7220 FOR R=0 TO 1
7230 PRINT SPC(21)"▒           "
7250 RETURN
```

El tercer apartado añade al dibujo el barrote transversal que une las dos primeras partes. El bucle **FOR...NEXT** hace que decrezca la variable **R** desde 2 hasta 0, de uno en uno. Se dibuja un paralelogramo, desde la parte alta del barrote transversal hasta que se rebasa el barrote vertical de la horca. El triángulo usado para construir cada mitad del paralelogramo se consigue pulsando **SHIFT** y la tecla **"£"**. La línea 7350 añade el último triángulo.

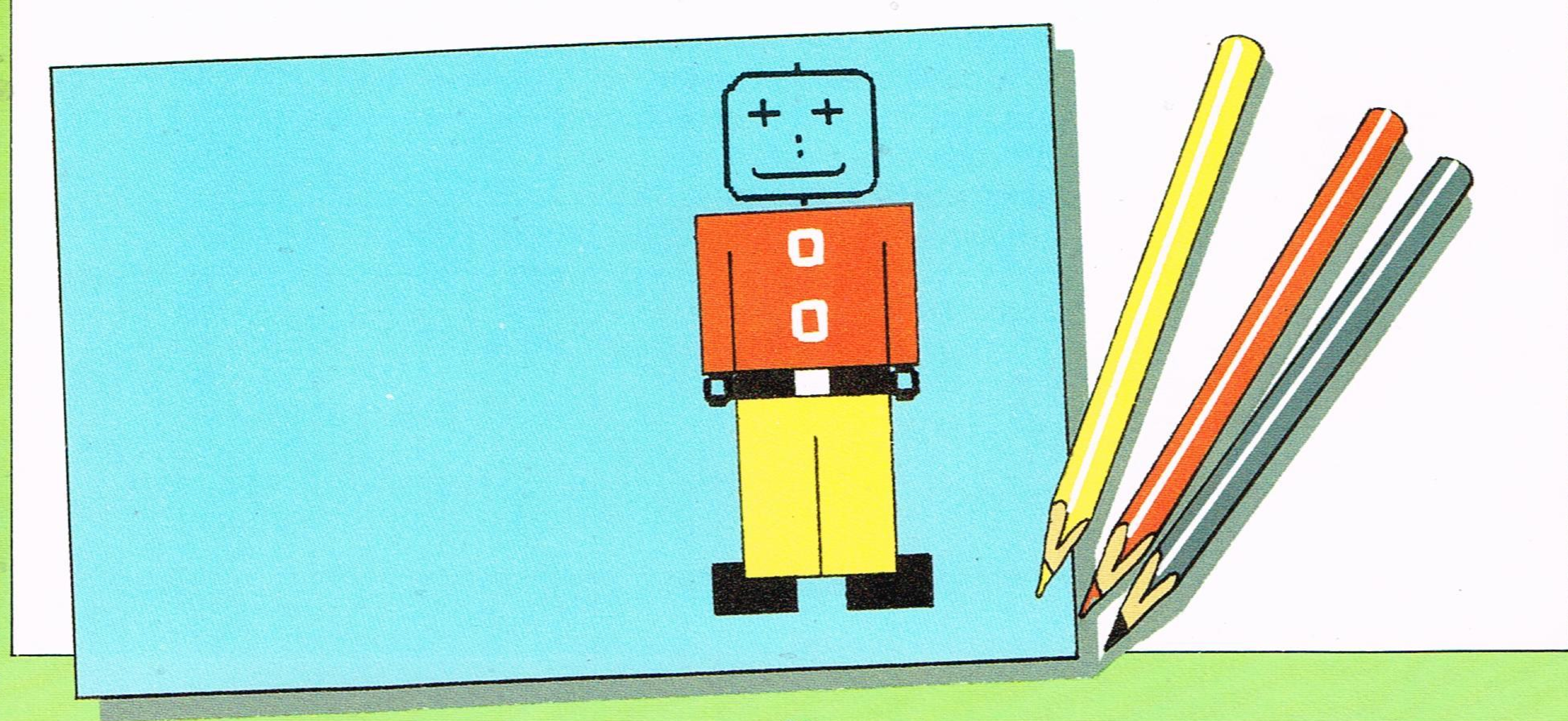
```
7300 REM PATIBULO 3
7310 PRINT"▒▒▒▒"
7320 FOR R=2 TO 0 STEP-1
7330 PRINT SPC(21+R)"▒▒▒▒"
7340 NEXT
7350 PRINTSPC(21)"▒"
7360 RETURN
```

```
7400 REM CUERDA
7410 PRINT "[illegible]"
7420 FOR R=0 TO 2: PRINT SPC(29)"|": NEXT
7430 RETURN
```

El cuarto apartado añade la cuerda. Esto se realiza con un gráfico sobre otro, en modo minúsculas, usando un bucle **FOR...NEXT.** Este bucle se repite una vez y la línea aparece en la columna vigésimonovena. Si ejecutas el programa obtendrás la horca completa en la pantalla.

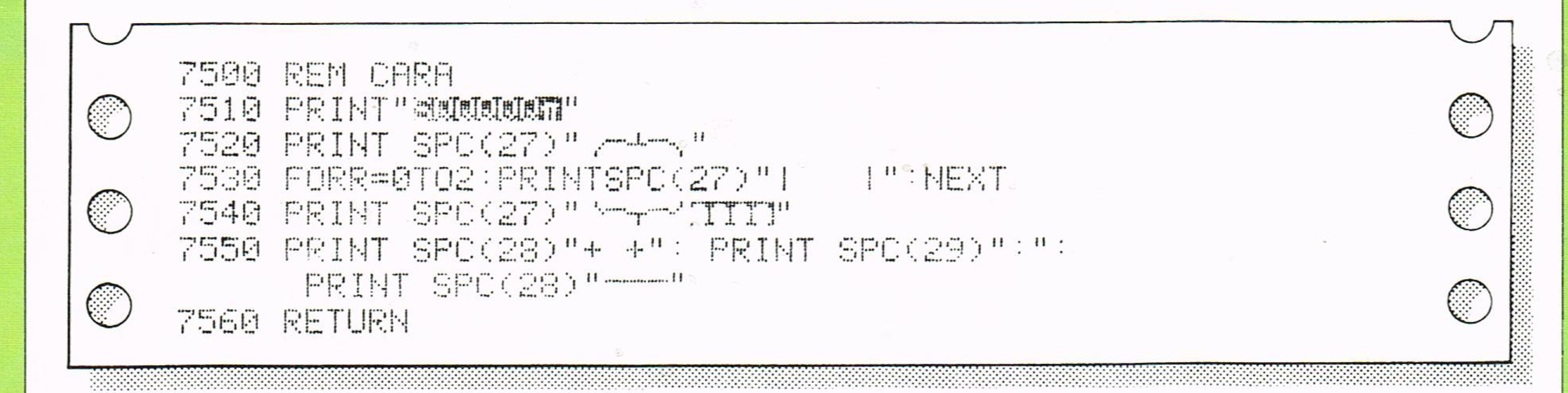


La siguiente sección gráfica dibujará el hombre. Antes de comenzar, sería una buena idea planificar el dibujo como se hace en el cuadro de abajo. En este momento puedes decidir qué colores vas a utilizar. Hay 16 colores disponibles en el Commodore 64, incluido el blanco y el negro. El gráfico del hombre se levanta en seis etapas.



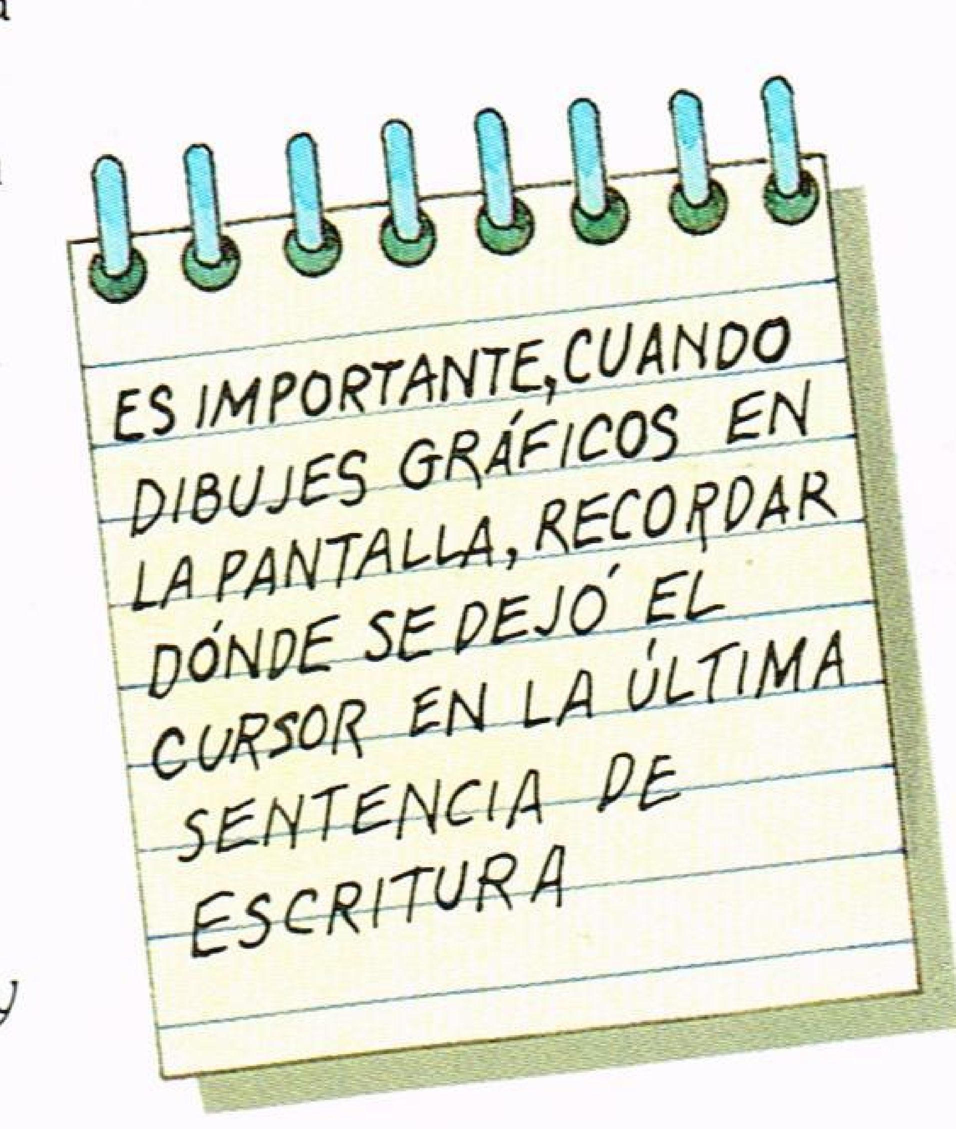
La primera parte del hombre es la cara. Recordarás que hemos elegido mover el cursor a través de la pantalla, en lugar de escribir la posición exacta y los colores usando **POKE**. Para esto necesitaremos programar tanto los mapas de pantalla como los de memoria del color.

La parte alta de la cara se dibuja usando los gráficos de las teclas U, E e I. Las mejillas se dibujan usando un bucle **FOR...NEXT** con el teclado en modo minúsculas (recuerda cómo hiciste la cuerda). Los gráficos de la parte baja de la cara se realizan con las teclas J, R̄ y K.



```
7500 REM CARA
7510 PRINT""
7520 PRINT SPC(27)"╭─┴─╮"
7530 FORR=0TO2:PRINTSPC(27)"|   |":NEXT
7540 PRINT SPC(27)"╰─┬─╯"
7550 PRINT SPC(28)"+ +": PRINT SPC(29)":":
      PRINT SPC(28)"──"
7560 RETURN
```

Los detalles de la cara se crean en el programa mediante una línea (la 7550), que coloca dos signos «más» como ojos, dos puntos como nariz y una línea como boca. Para colocarlos en su lugar se debe mover el cursor hacia arriba cuatro líneas; esto se realiza al final de la línea 7540. Si una sentencia de escritura termina con un punto y coma, o dos puntos, entonces el cursor se moverá una línea hacia abajo. Por eso, en la línea 7550, los rasgos de la cara aparecen unos debajo de otros, aunque las instrucciones estén en la misma línea.



La siguiente subrutina construye el cuerpo. El color (rojo claro) y la posición se dan en la línea 7610. El bucle **FOR...NEXT** dibuja un rectángulo de tres bloques de ancho y cinco líneas de largo, de la misma forma en que se dibujó la horca. La línea 7630 sube el cursor cuatro líneas y dibuja una "O" en modo inverso. Otra "O" igual se dibuja dos filas más abajo que la anterior. Estas "oes" simulan ser los botones de la chaqueta del ahorcado. Si quieres puedes utilizar una instrucción parecida para dotar al ahorcado de bolsillos usando dos símbolos de sustracción. Sería una buena idea, llegado a este punto ejecutar el programa para comprobar que has colocado el cuerpo correctamente.

```
7600 REM CUERPO
7610 PRINT""
7620 FORR=0TO4:PRINT SPC(28)"   ":NEXT
7630 PRINT""SPC(29)"O":PRINT SPC(29)"O"
7640 RETURN
```

Por la siguiente vida perdida se añade un brazo al ahorcado. El brazo se dibuja en tres pasos, hombro, brazo y mano. El hombro se realiza con dos bloques (espacios en modo inverso) en la línea 7720. Para separar el brazo del cuerpo se necesita un bloque con un fragmento de espacio colocado en el lado derecho. N̄ produce una línea en la derecha, y al escribirlo en modo inverso se forma el espacio requerido. La línea 7730 es un bucle de tres pasadas que dibuja el brazo. La mano se representa por una "O".

```
7700 REM BRAZO 1
7710 PRINT"[illegible]"
7720 FORR=0TO1:PRINT SPC(27)"[illegible] ": NEXT
7730 FORR=0TO2: PRINT SPC(27)"[illegible]|":NEXT
7740 PRINT SPC(27)"O"
7750 RETURN
```

La siguiente vida perdida produce el segundo brazo, de la misma manera que el primero. Sólo cambia el número entre paréntesis de **SPC( ),** para colocar el brazo en el lado derecho del cuerpo. La H̄ se usa para separar este brazo.

```
7800 REM BRAZO 2
7810 PRINT"[illegible]"
7820 FORR=0TO1:PRINT SPC(31)"[illegible] ":NEXT
7830 FORR=0TO2:PRINT SPC(31)"[illegible] ":NEXT
7840 PRINT SPC(31)"O"
7850 RETURN
```

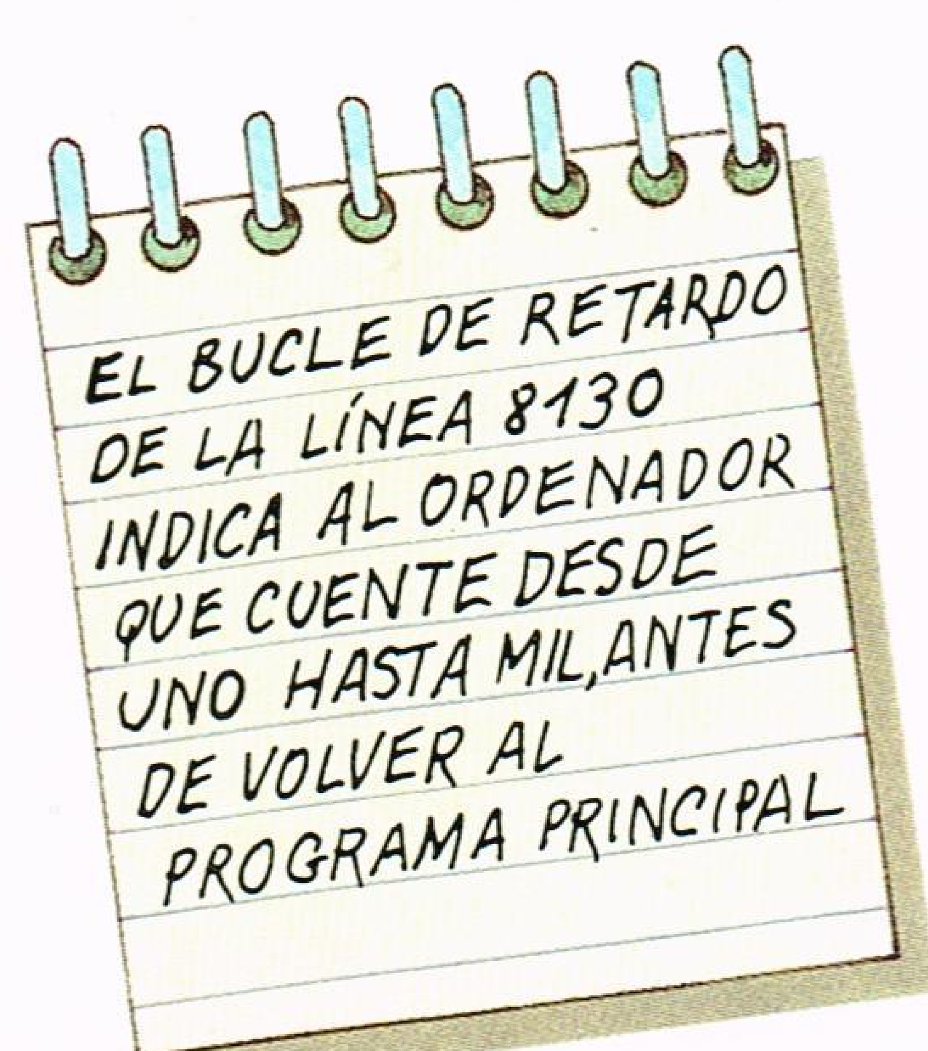
La siguiente parte del ahorcado que se añade son las piernas. La primera pierna se crea por medio de un bucle de seis pasadas que usa los espacios **REVERSE ON** y K̄ **REVERSE OFF**; ésta aparecerá cuando se pierda la novena vida. Cada una ocupará un espacio y medio —el cuerpo se ha construido con tres espacios y cada pierna ha de ser la mitad del cuerpo—. La instrucción dada dibuja las piernas en verde claro, pero, por supuesto, puedes dibujarlas en el color que quieras con solo usar otro de los códigos de color. Las piernas se dibujan comenzando en la línea quince. La línea 7910 podría establecerse como **PRINT LEFT$(UD$,16)** para colocar la pierna en la línea dieciséis. Esto se haría para evitar el tener que escribir y contar un número muy largo de símbolos de bajar cursor que nos podrían llevar a error. El código del color tendrá que ser situado entre comillas.

```
7900 REM PIERNA 1
7910 PRINT"[illegible]|"
7920 FORR=0TO5: PRINT SPC(28)"[illegible] [illegible]|":NEXT
7930 RETURN
```

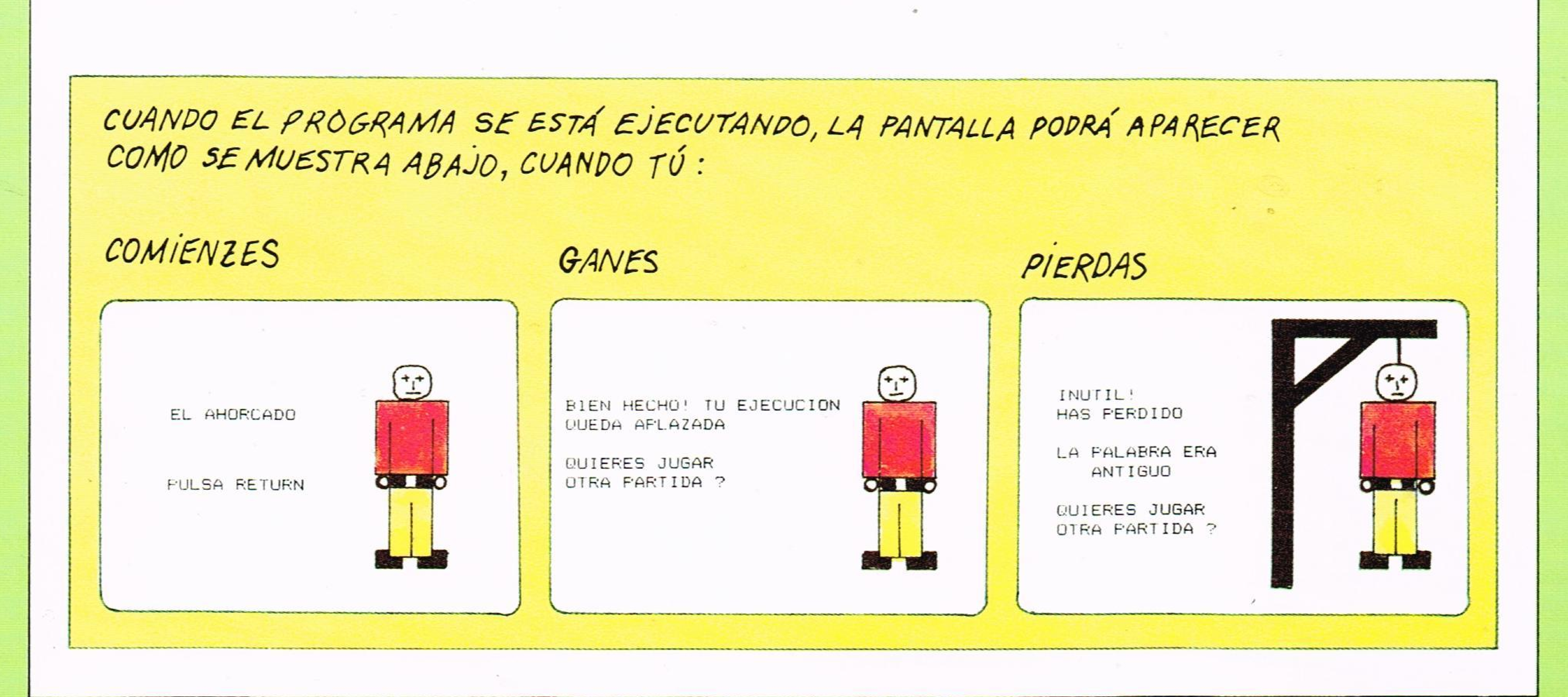
```
8000 REM PIERNA 2
8010 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
8020 FORR=0TO5:PRINT SPC(28)"▒ |  ":NEXT
8030 PRINT SPC(28)"▒▒▒▒▒▒▒▒▒ * "
8040 PRINT SPC(28)"▒▒▒▒▒▒▒▒¯ ¯"
8120 REM RETARDO
8130 FORR=1TO1000:NEXT R
8140 RETURN
```

La segunda pierna se dibuja de la misma forma que la primera. Las piernas se dibujan como un largo bloque con una línea en el medio, la cual se produce usando las teclas en modo minúsculas. Recuerda, la línea 8020 está toda en inverso, para que aparezca como un hueco.

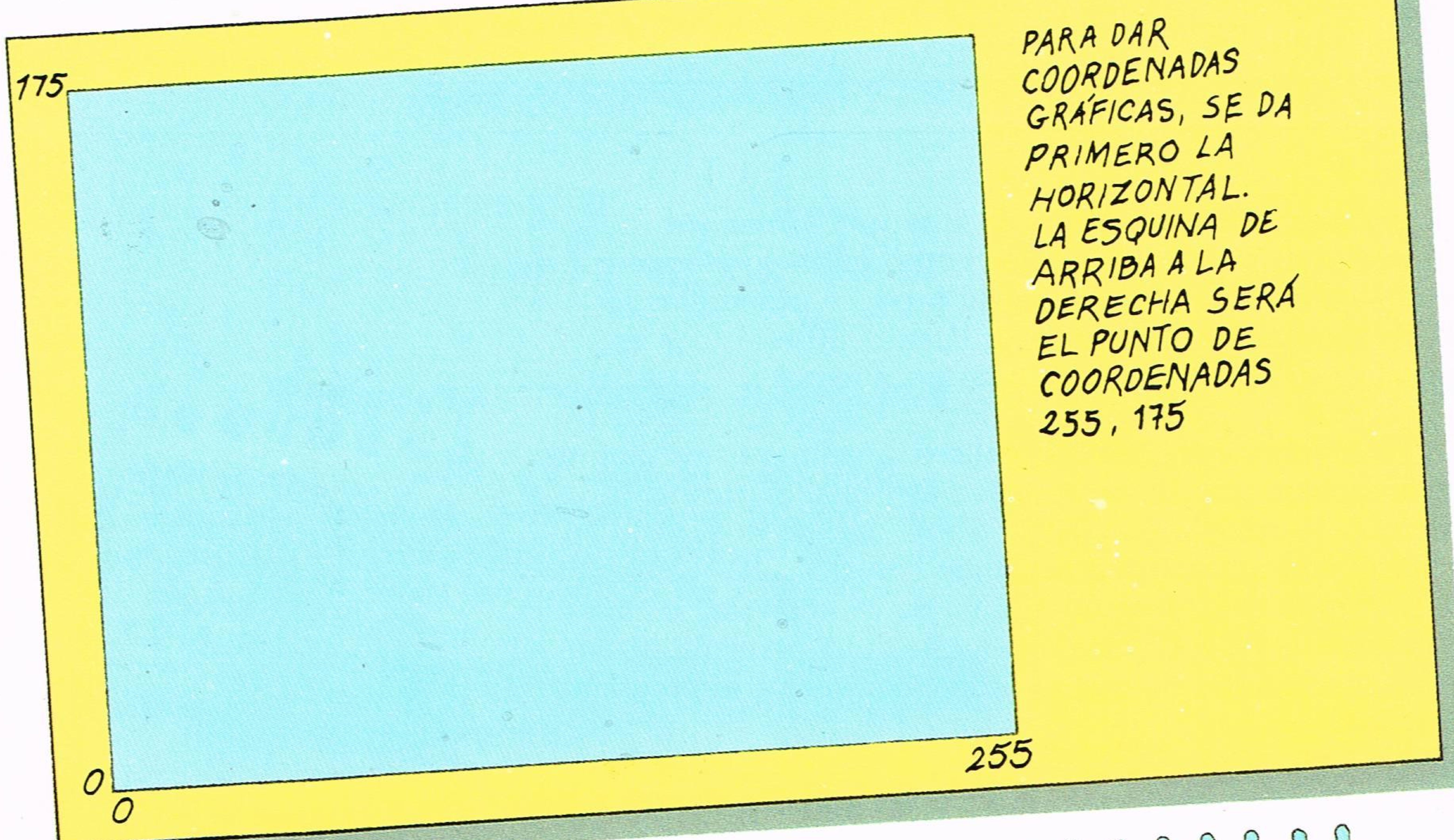
En esta última etapa de los gráficos, añadiremos un cinturón y unos zapatos para dotar de más detalles al ahorcado. El cinturón se colorea en gris; se dibuja usando el espacio en inverso, con un asterisco actuando como hebilla. Los zapatos se pintan de color marrón al final de cada pierna, y para ello utilizamos la tecla U. Finalmente, hay una pequeña parada en esta sección antes de que el ordenador vuelva al programa principal.



Con esta última sección, el programa ya está completo y listo para ejecutarse. Los gráficos del ahorcado son llamados desde la subrutina de presentación para que el ahorcado aparezca en tu pantalla cuando escribas **RUN**. Se llama también a los gráficos cuando se gana o pierde el juego. Al final del libro se da un listado del programa entero, y hay algunas adiciones para hacer el programa más interesante. Hay ruidos y movimientos de boca; por ejemplo, cuando ganas el ahorcado sonríe y cuando pierdes su boca se tuerce hacia abajo.



La pantalla de textos del Spectrum se divide en 22 líneas y 32 columnas verticales. Para comandos gráficos se usa otra división con 255 posiciones horizontales y 175 verticales. Cuando damos coordenadas gráficas, la posición horizontal se da primero.

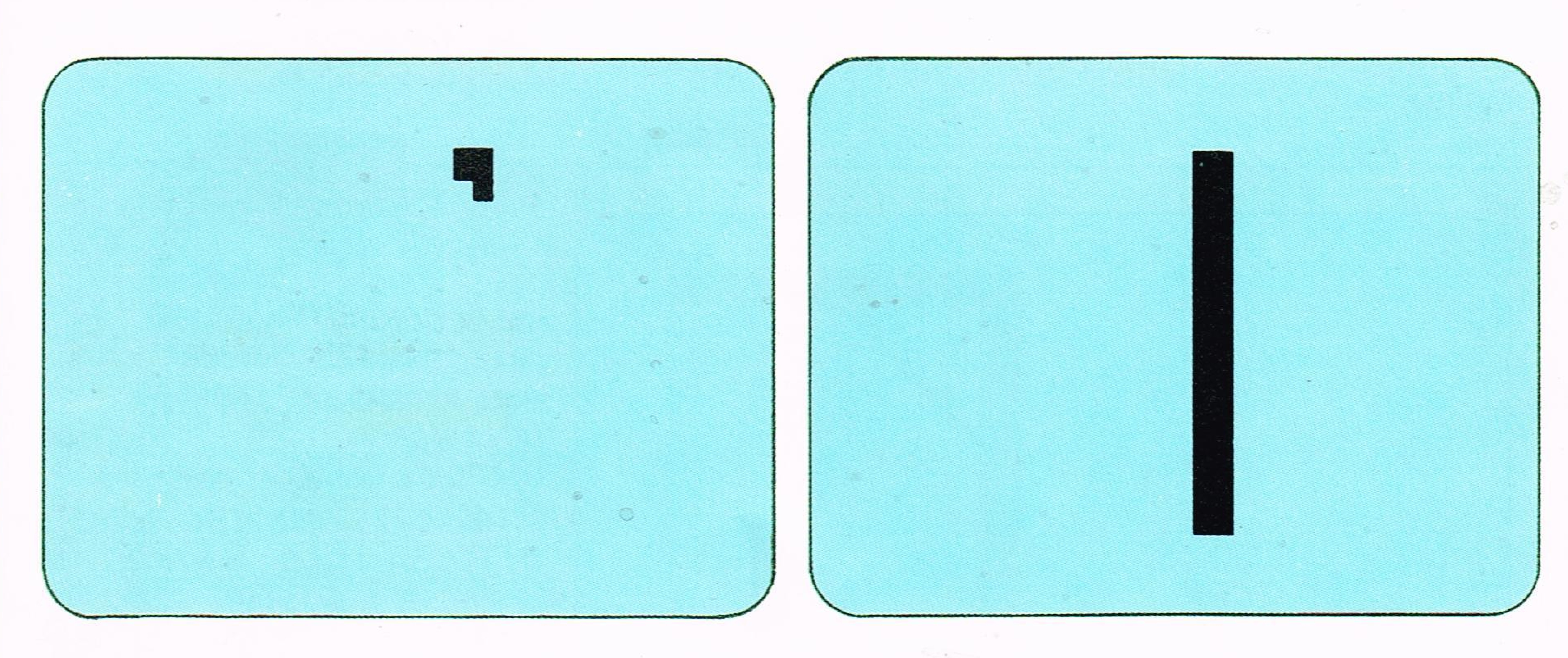


La subrutina del *ahorcado* se divide en diez etapas; cada etapa comienza cuando se ha hecho un intento fallido y se ha perdido una vida. En la línea 7010 el valor de **vida** se incrementa en uno cada vez que se llama a la rutina, y se comprueba si dicho valor ha sobrepasado el valor 10 (en tal caso el juego se ha perdido). La línea 7020 dirige el programa a la sección correspondiente del gráfico, usando el nuevo valor de **vida.** La primera sección del gráfico construye los postes de la horca, usando un simple bloque gráfico representado por **CHR$143** y dos bucles **FOR...NEXT,** que dan diferentes valores de columnas y líneas a la instrucción **PRINT AT.**

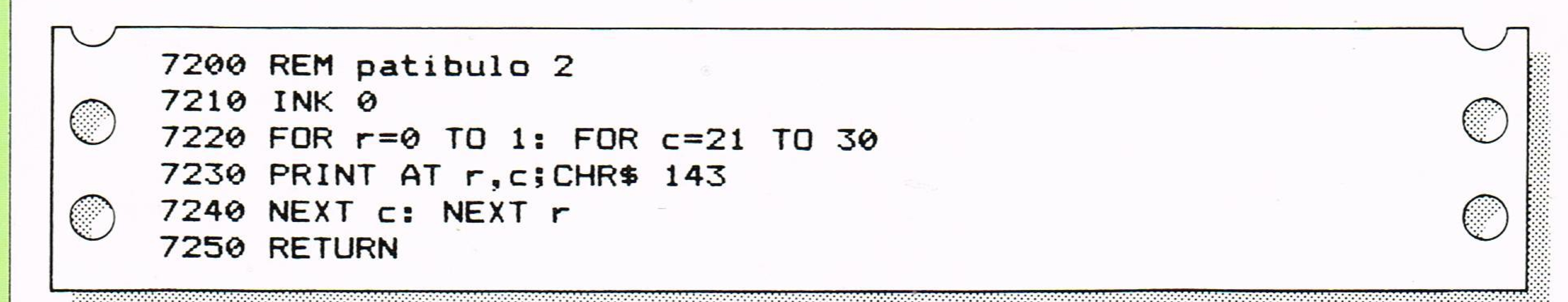
UN BUCLE DENTRO DE OTRO BUCLE SE LLAMA BUCLE ANIDADO. EL SEGUNDO BUCLE (FOR C=19 TO 20) DEBE COMPLETARSE PRIMERO.

```
7000 REM hombre ahorcado
7010 LET vida=vida+1: IF vida=10 THEN  LET pierdes=1
7020 GO TO 7000+vida*100
7100 REM patibulo 1
7110 INK 0
7120 FOR r=0 TO 21: FOR c=19 TO 20
7130 PRINT AT r,c;CHR$ 143
7140 NEXT c: NEXT r
7150 RETURN
```

CHR$143 es el código de un bloque, en este caso coloreado en negro por la instrucción **INK 0.** El bucle **FOR...NEXT** hace que el valor de la fila varíe desde 0 hasta 21 en las columnas 19 y 20, para marcar más el poste. Para probar esta primera sección de gráficos teclea **GOSUB 7100.** En lugar de usar los códigos **CHR$,** puedes utilizar los símbolos gráficos disponibles directamente en la fila superior del Spectrum cuando éste está en modo gráfico.



El segundo paso consiste en añadir la barra horizontal de la horca. Para ello se usa el mismo método que para la construcción del poste.



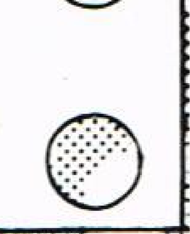
```
7200 REM patibulo 2
7210 INK 0
7220 FOR r=0 TO 1: FOR c=21 TO 30
7230 PRINT AT r,c;CHR$ 143
7240 NEXT c: NEXT r
7250 RETURN
```

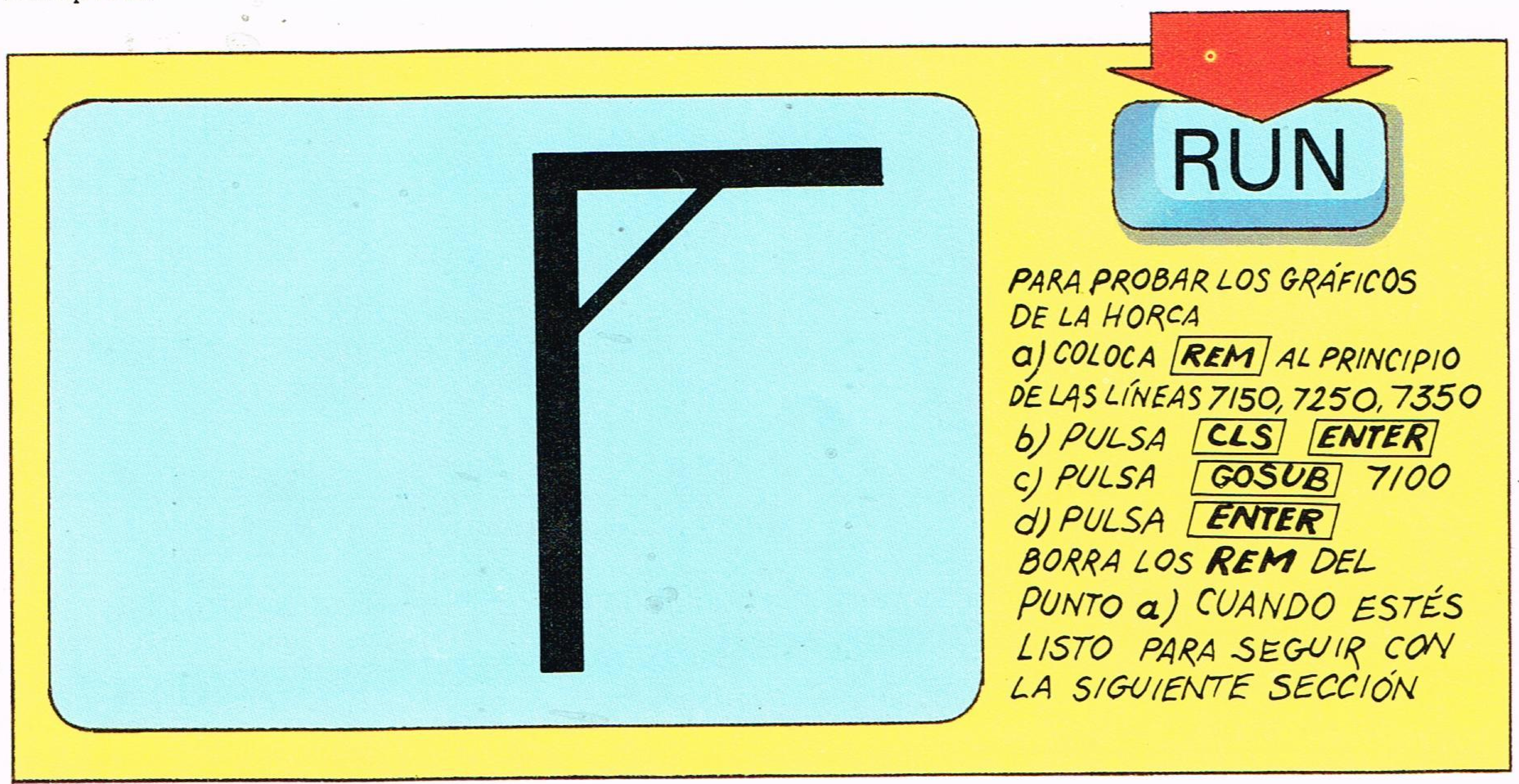
La tercera etapa es colocar la barra transversal. Para ello, con las dos órdenes gráficas **PLOT** y **DRAW,** se dibujan una serie de líneas diagonales. La instrucción **PLOT** indica las coordenadas del punto donde comenzará la línea. La instrucción **DRAW** dibuja una línea desde el punto anterior hasta el punto resultante de sumar a la primera coordenada del punto inicial el primer número que sigue a **DRAW,** y a la segunda coordenada, el segundo.

```
7300 REM patibulo 3
7310 INK 0
7320 FOR r=100 TO 105
7330 PLOT 155,r: DRAW 60,60
7340 NEXT r
```

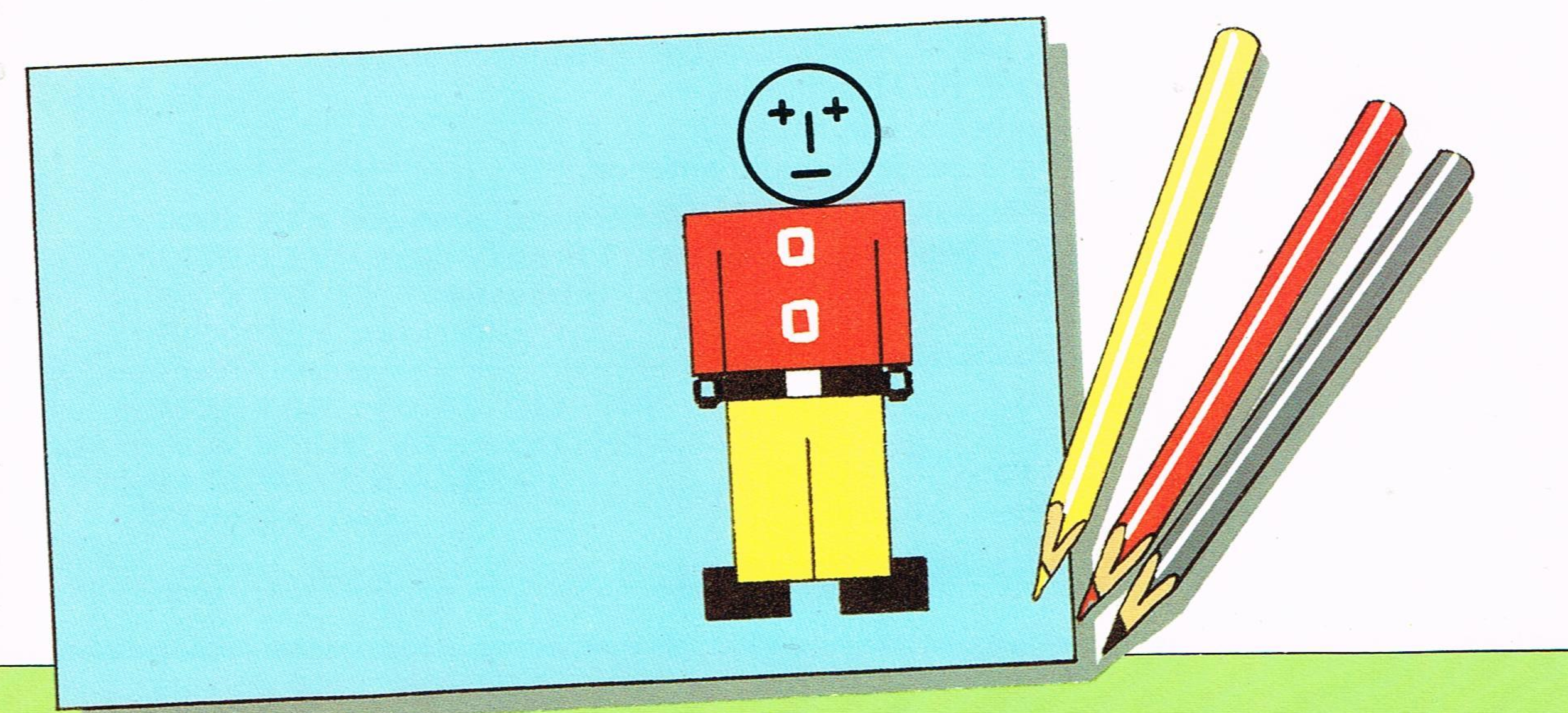
```
7400 REM cuerda
7410 INK 0: PLOT 227,135: DRAW 0,25
7420 RETURN
```

El cuarto paso consiste en añadir la cuerda. Otra vez se usan las instrucciones **PLOT** y **DRAW.** Las coordenadas dadas aquí proporcionan a la cuerda una longitud adecuada, de forma que quede debajo suficiente espacio para el hombre. Ejecuta el programa como se indica más abajo y verás la horca completa.





La sección siguiente dibujará el hombre. Antes de empezar, convendría planear tu dibujo, como en la figura de abajo. En este punto puedes elegir los colores que va a llevar; hay ocho opciones en el Spectrum, incluido el negro y el blanco. El dibujo del hombre se construye en seis etapas.



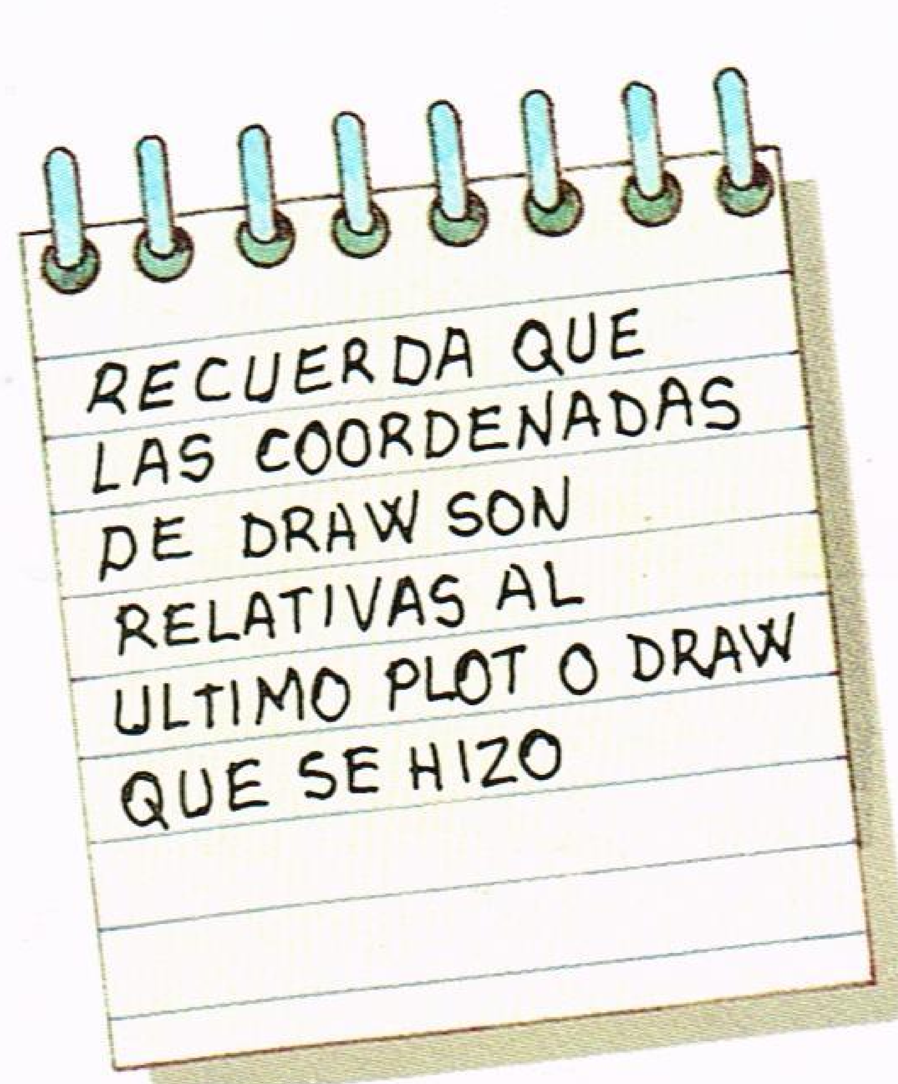
SPECTRUM

La primera parte del hombre es la cara. El Spectrum tiene una instrucción llamada **CIRCLE** para dibujar círculos. Los dos primeros números que la acompañan indican la posición del centro del círculo, y el tercero su radio.

La línea 7520 dibuja dos signos "más" (+) para hacer los ojos del hombre. Observa que para situarlos se usan las coordenadas de la rejilla de textos. Para dibujar la nariz y la boca se utilizan las instrucciones **PLOT** y **DRAW**, respectivamete.

```
7500 REM cara
7510 INK 0: CIRCLE 227,119,15
7520 PRINT AT 6,27;"+ +"
7530 PLOT 227,115: DRAW 0,7
7540 PLOT 222,110: DRAW 10,0
7550 RETURN
```

La siguiente parte del programa se encarga de dibujar el cuerpo del ahorcado. Esto se podría hacer dibujando una serie de bloques de colores, como se hizo para dibujar la horca en las secciones anteriores. Un método alternativo es usar la instrucción **PAPER** para colorear el fondo de la pantalla. Por tanto, se puede indicar al ordenador que imprima una serie de espacios coloreados como se desee y añadir luego algunos detalles extra en esos espacios.



En la parte del programa que se muestra abajo, la instrucción **PAPER 2** da un color rojo al fondo. Los espacios coloreados que forman el cuerpo se sitúan desde la fila 9 hasta la 13; como se imprimen directamente, se usa la rejilla de texto. Resultan suficientes tres espacios, comenzando por la columna 27. La línea 7630 hace que se dibujen dos "oes" (coloreadas en blanco por la instrucción **INK 7**) en medio del cuerpo, representando los botones de la chaqueta del ahorcado. Si quieres puedes usar las órdenes **PLOT** y **DRAW** para dotar de bolsillos al dibujo, o hacerlo con signos "menos" (−). Sería una buena idea, alcanzado este punto, ejecutar el programa para comprobar que el dibujo está colocado correctamente y no ocupa la posición de la izquierda de la pantalla, reservada para los mensajes.

```
7600 REM cuerpo
7610 PAPER 2
7620 FOR r=9 TO 13: PRINT AT r,27;"   ": NEXT r
7630 INK 7:PRINT AT 10,28;"O":PRINT AT 12,28;"O"
7640 RETURN
```

SPECTRUM

En la siguiente vida perdida se le añade al ahorcado un brazo. El brazo se dibuja de la misma forma que el cuerpo: se colorea la pantalla de rojo y se imprimen espacios en la columna 26, usando un bucle **FOR...NEXT** para utilizar la fila adecuada. Para separar los brazos del cuerpo se dibuja una línea negra con las instrucciones **PLOT** y **DRAW** de la línea 7730. La línea 7740 cambia a blanco el color del papel, y la instrucción **CIRCLE** se usa para dibujar un pequeño círculo al final del brazo, que representa la mano del hombre.

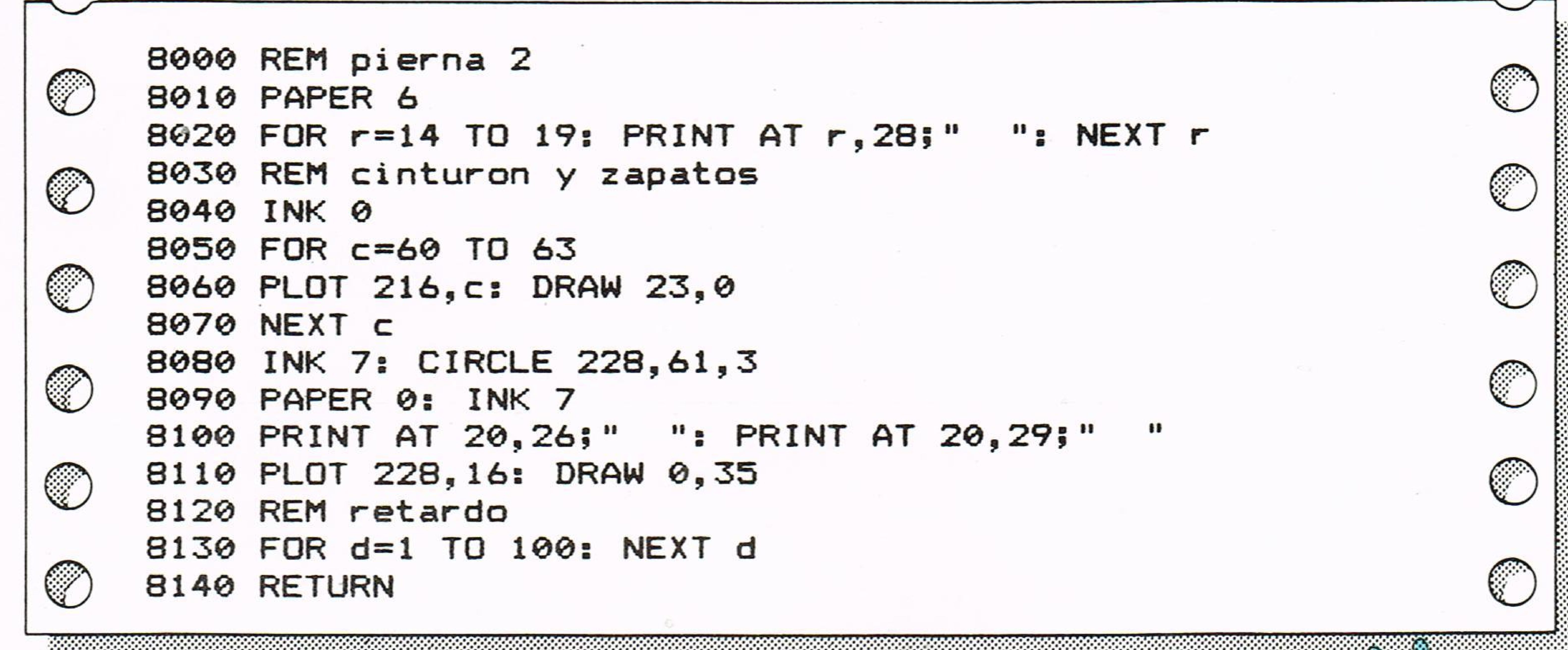
```
7700 REM brazo 1
7710 PAPER 2: INK 7
7720 FOR r=9 TO 14: PRINT AT r,26;" ": NEXT r
7730 PLOT 215,56: DRAW 0,35
7740 PAPER 7: INK 0: CIRCLE 210,50,4
7750 RETURN
```

La siguiente vida perdida produce el siguiente brazo. Se hace exactamente igual que el primero; sólo cambian las coordenadas de la columna, para situar el brazo en el lado derecho del cuerpo.

```
7800 REM brazo 2
7810 PAPER 2: INK 7
7820 FOR r=9 TO 14: PRINT AT r,30;" ": NEXT r
7830 PLOT 240,56: DRAW 0,35
7840 PAPER 7: INK 0: CIRCLE 244,50,4
7850 RETURN
```

El siguiente gráfico que se añade al ahorcado es la primera pierna. Se construye con dos bloques gráficos, correspondientes a los códigos **CHR$ 128** y **CHR$ 133.** **CHR$ 128** es un espacio vacío, coloreado de amarillo por la instrucción **PAPER 6** de la línea 7910. **CHR$ 133** es un carácter con dos partes. La mitad izquierda toma el color del papel y la derecha el dado por la instrucción **INK.** En nuestro caso, el papel da el color amarillo a una mitad del bloque mientras que **INK 7** (que es el color blanco) hace que el lado derecho del carácter se confunda con el color blanco del fondo de la pantalla. El efecto resultante es una pierna amarilla de bloque y medio de anchura.

```
7900 REM pierna 1
7910 PAPER 6: INK 7
7920 FOR r=14 TO 19: PRINT AT r,27;CHR$ 128;CHR$ 133: NEXT r
7930 RETURN
```



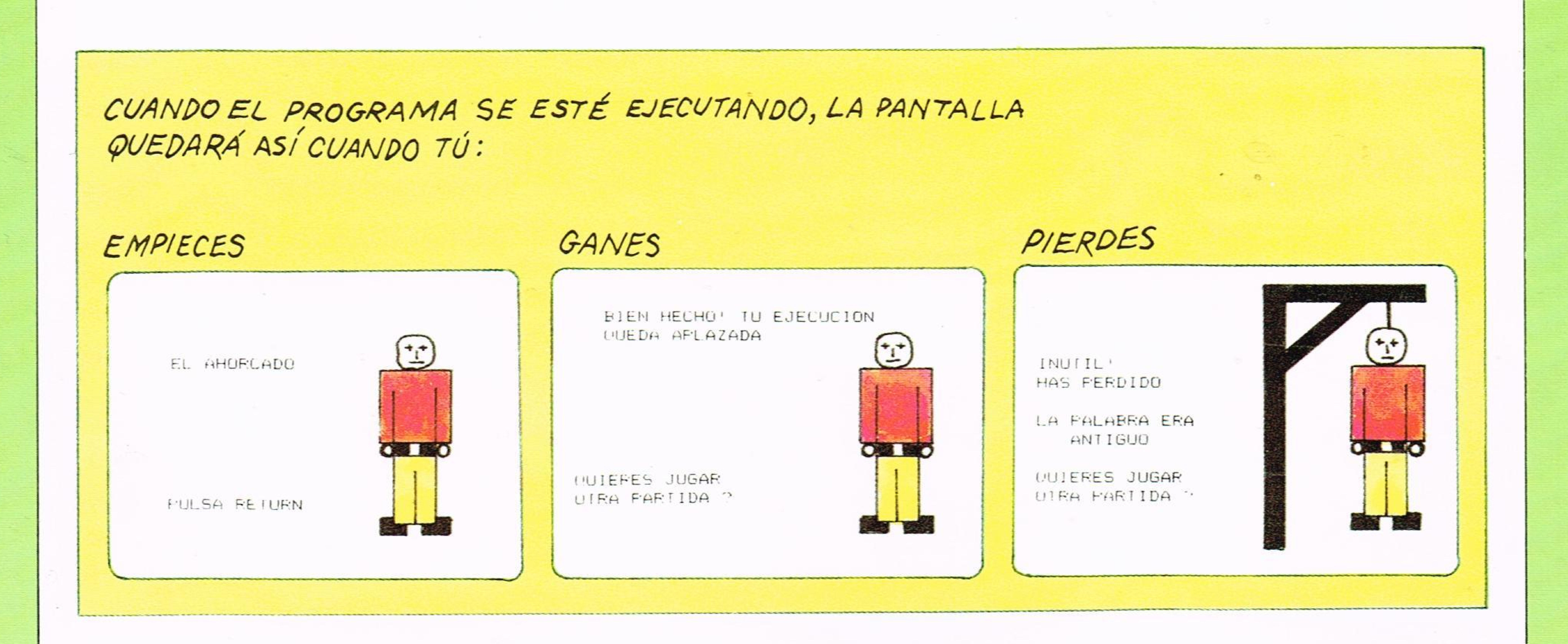
```
8000 REM pierna 2
8010 PAPER 6
8020 FOR r=14 TO 19: PRINT AT r,28;"  ": NEXT r
8030 REM cinturon y zapatos
8040 INK 0
8050 FOR c=60 TO 63
8060 PLOT 216,c: DRAW 23,0
8070 NEXT c
8080 INK 7: CIRCLE 228,61,3
8090 PAPER 0: INK 7
8100 PRINT AT 20,26;"  ": PRINT AT 20,29;"  "
8110 PLOT 228,16: DRAW 0,35
8120 REM retardo
8130 FOR d=1 TO 100: NEXT d
8140 RETURN
```

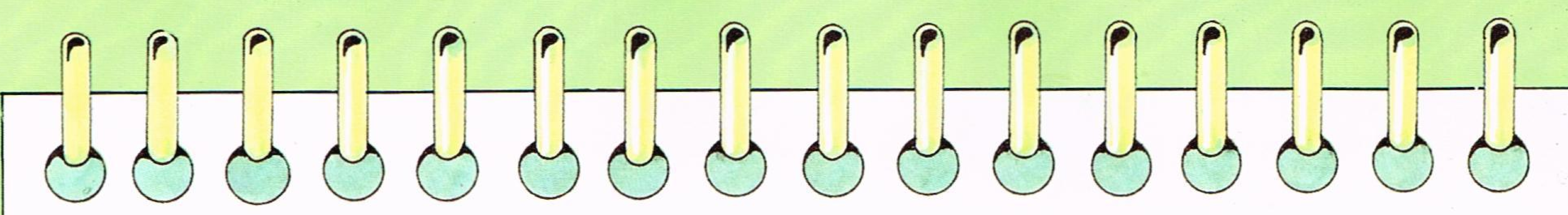
La segunda pierna se dibuja imprimiendo una serie de espacios coloreados, de dos bloques de anchura, pintando el medio bloque de **CHR$ 133** usado para construir la primera pierna. En este caso, **PAPER 6** da un color amarillo al fondo. En esta última etapa del gráfico añadiremos un cinturón y unos zapatos para proporcionar más detalles al dibujo. El cinturón se construye con cuatro líneas negras, usando para ello **PLOT, DRAW** y un sencillo bucle **FOR...NEXT.** En la línea 8080, la instrucción **CIRCLE** dibuja un pequeño círculo que simula ser la hebilla. Los zapatos se dibujan con espacios en negro al final de cada pierna. Por último, hay un pequeño retardo antes de que el ordenador vuelva al programa principal.

Con esta última sección, el programa ya está listo para ser ejecutado **(RUN).** Como la rutina de presentación llama a la de los gráficos, el ahorcado aparecerá en la pantalla cuando pulses **RUN.** El dibujo también aparecerá cuando el juego se gane o se pierda. Al final del libro tienes un listado completo del programa.

EL BUCLE DE RETARDO DE LA LÍNEA 8130 INDICA AL ORDENADOR QUE CUENTE DE UNO A CIEN, ANTES DE VOLVER AL PROGRAMA PRINCIPAL

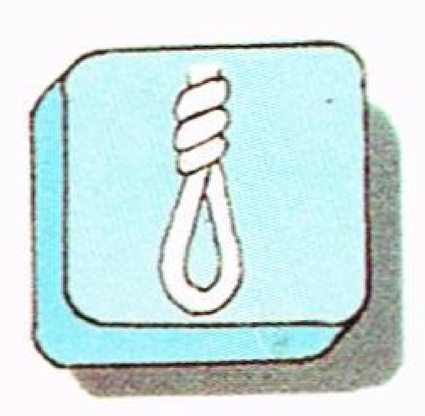
CUANDO EL PROGRAMA SE ESTÉ EJECUTANDO, LA PANTALLA QUEDARÁ ASÍ CUANDO TÚ:





Mejora tus programas (2)

Ahora que tienes escrito el programa completo, puedes salvarlo en cinta (el método para almacenar programas se da en el manual del usuario). ¿Por qué no pruebas gráficos diferentes? Tal vez puedas construir un edificio.



Listado completo

El programa listado abajo corresponde al juego del ahorcado. Las líneas marcadas con un asterisco (*) son adicionales al programa presentado a lo largo del libro. Por ejemplo, cuando ganas el ahorcado sonríe, y cuando pierdes sus ojos se cierran y sus pies giran hacia abajo.

El programa de la pág. 42 es el juego del ahorcado para el Spectrum. Hay algunas líneas adicionales marcadas con asterisco (*) que no han sido expuestas a lo largo del libro. Dan detalles adicionales del juego y poseen números de línea diferentes para que puedas escribir el programa sin ellas.

COMMODORE

```
  10 PRINT"■"
  20 GOSUB 1000:REM SELECCIONA PALABRA
  30 GOSUB 2000:REM TITULO
  40 IF GA=1 OR PI=1THEN 80
  50 GOSUB 3000:REM ELIGE LETRA
  60 IF CO=0 THEN GOSUB 7000: REM HOMBRE AHORCADO
  70 GOTO 40
  80 IF GA =1 THEN GOSUB 4000: REM GANAR
  90 IF PI =1 THEN GOSUB 5000: REM PERDER
 100 GOSUB 6000:REM OTRA PARTIDA
 110 STOP
1000 REM SELECCIONA PALABRA
1005 UD$ ="■■■■■■■■■■■■■■■■■■■■■■■■■"
1010 P =8
1020 RESTORE
1030 EL =INT(RND(0)*P)+1
1040 FOR L =1TO EL
1050 READ P$
1060 NEXT L
1070 LO =LEN(P$):D$=P$
1080 RETURN
2000 REM TITULO
2010 PRINT"■"
2020 FOR L=4 TO 10
2030 ON L-4 GOSUB 7500,7600,7700,7800,7900,8000
2040 NEXTL
2050 PRINT LEFT$(UD$,10)SPC(12)"■AHORCADO"
2060 INPUT"■■■■■■■■■■■■■■         PULSA RETURN";A$
2070 PRINT"■■■■■■LETRAS ESCOGIDAS"
2080 PRINT"■■■--------------------"
2090 PRINT"■■■■■■■■■■■■■■LA PALABRA"
2100 PRINT"■■■DEL ORDENADOR ES"
2110 PRINT"■■■■■■";
2120 FOR L =1 TO LO:PRINT"*";:NEXT L :PRINT
2130 PRINT"■■■■POR FAVOR, ESCOGE"
2140 PRINT"■■■■■UNA LETRA"
```

```
 2150 GA=0:PI=0:VI=0:PU=0
 2160 RETURN
 3000 REM ELIGE LETRA
 3010 PRINT LEFT$(UD$,23)"█    ████";
 3020 INPUT C$:C$=LEFT$(C$,1)
 3030 PRINT"█████████"SPC(VI)C$
 3040 CO=0
 3050 FOR P=1 TO LO
 3060 IF C$<>MID$(P$,P,1) THEN 3100
 3070 CO=1:PU=PU+1
 3080 P$=LEFT$(P$,P-1)+"?"+RIGHT$(P$,LO-P)
 3090 PRINTLEFT$(UD$,18)"██████"SPC(P-1)C$
 3100 NEXT P
 3110 IF PU=LO THEN GA=1
 3120 RETURN
 4000 REM GANAR
 4010 PRINT"J"
 4020 FOR L = 4 TO 10
 4030 ON L-4 GOSUB 7500,7600,7700,7800,7900,8000
 4040 NEXT L
*4045 PRINT "████████████"SPC(28)"\—/"
 4050 PRINT LEFT$(UD$,7)"  █BIEN HECHO ! TU EJECUCION"
 4060 PRINT "  QUEDA APLAZADA"
*4063 FOR I=0 TO 24:POKE 54272,0:NEXT I
*4067 POKE 54277,15:POKE 54278,240:POKE 54296,15
*4068 POKE 54276,33:FOR I=0 TO 255:POKE 54276,I: NEXT
*4069 POKE 54276,32
 4070 RETURN
 5000 REM PERDER
 5010 PRINT"J"
 5020 FOR L=1 TO 10
 5030 ON L GOSUB 7100,7200,7300,7400,7500,7600,7700,7800,7900,8000
 5040 NEXT
 5050 PRINT "█████████  INUTIL !"
 5060 PRINT"█     HAS PERDIDO !"
 5070 PRINT"█████LA PALABRA ERA█": PRINTSPC(5)D$
*5072 FOR I=0 TO 24:POKE 54272,0:NEXT
*5074 POKE 54277,15:POKE 54278,240: POKE 54296,15
*5076 POKE 54276,33:FOR I=255 TO 0 STEP-1:POKE 54273,I: NEXT
*5077 POKE 54276,32
*5078 PRINT"████████████"SPC(28)"/—\"
*5079 PRINT"██████████████"SPC(28)"██ █ "
 5080 RETURN
 6000 REM OTRA PARTIDA
 6010 PRINT"█████████████████████  QUIERES JUGAR"
 6020 PRINT"   OTRA PARTIDA";
 6030 INPUTA$
 6040 IF LEFT$(A$,1)="S" THEN 10
 6050 PRINT"████████████   ADIOS"
 6060 RETURN
 7000 REM HOMBRE AHORCADO
 7010 VI=VI+1:IF VI=10 THEN PI=1
 7020 ON VI GOTO 7100,7200,7300,7400,7500,7600,7700,7800,7900,8000
 7100 REM PATIBULO 1
 7110 PRINT"███"
 7120 FOR R=0 TO 21
 7130 PRINT SPC(19)"█  "
 7140 NEXT
 7150 RETURN
 7200 REM PATIBULO 2
 7210 PRINT"███"
 7220 FOR R=0 TO 1
 7230 PRINT SPC(21)"█           "
 7250 RETURN
 7300 REM PATIBULO 3
```

```
7310 PRINT"■■■■"
7320 FOR R=2 TO 0 STEP-1
7330 PRINT SPC(21+R)"■■■"
7340 NEXT
7350 PRINTSPC(21)"■"
7360 RETURN
7400 REM CUERDA
7410 PRINT "■■■■"
7420 FOR R=0 TO 2: PRINT SPC(29)"!": NEXT
7430 RETURN
7500 REM CARA
7510 PRINT"■■■■■■■"
7520 PRINT SPC(27)" ╭─┴─╮"
7530 FORR=0TO2:PRINTSPC(27)"|    |":NEXT
7540 PRINT SPC(27)"╰─┬─╯TTTT"
7550 PRINT SPC(28)"+ +": PRINT SPC(29)":": PRINT SPC(28)"───"
7560 RETURN
7600 REM CUERPO
7610 PRINT"■■■■■■■■■■■■■"
7620 FORR=0TO4:PRINT SPC(28)"■   ":NEXT
7630 PRINT"TTTT"SPC(29)"■O":PRINT SPC(29)"■■O"
7640 RETURN
7700 REM BRAZO 1
7710 PRINT"■■■■■■■■■■■■■"
7720 FORR=0TO1:PRINT SPC(27)"■ ": NEXT
7730 FORR=0TO2: PRINT SPC(27)"■|":NEXT
7740 PRINT SPC(27)"O"
7750 RETURN
7800 REM BRAZO 2
7810 PRINT"■■■■■■■■■■■■■"
7820 FORR=0TO1:PRINT SPC(31)"■ ":NEXT
7830 FORR=0TO2:PRINT SPC(31)"■| ":NEXT
7840 PRINT SPC(31)"O"
7850 RETURN
7900 REM PIERNA 1
7910 PRINT"■■■■■■■■■■■■■■■■■■■■"
7920 FORR=0TO5: PRINT SPC(28)"■ ■|":NEXT
7930 RETURN
8000 REM PIERNA 2
8010 PRINT"■■■■■■■■■■■■■■■■■■■■"
8020 FORR=0TO5:PRINT SPC(28)"■ |  ":NEXT
8030 PRINT SPC(28)"TTTTTTTT■ * "
8040 PRINT SPC(28)"■■■■■■■ ‾ ‾"
8120 REM RETARDO
8130 FORR=1TO1000:NEXT R
8140 RETURN
9000 DATA MANZANA,ANGULO,ALFABETO,ANTIGUO
9010 DATA ALUBIAS,LIBRO,BUFALO,PLAYA
```

SPECTRUM

```
  10 CLS
  20 GO SUB 1000: REM selecciona palabra
  30 GO SUB 2000: REM titulo
  40 IF ganas OR pierdes=1 THEN  GO TO 80
  50 GO SUB 3000: REM elige letra
  60 IF correcto=0 THEN  GO SUB 7000: REM hombre ahorcado
  70 GO TO 40
  80 IF ganas=1 THEN  GO SUB 4000: REM ganar
  90 IF pierdes=1 THEN  GO SUB 5000: REM perder
 100 GO SUB 6000: REM otra partida ?
 110 STOP
1000 REM selecciona palabra
1010 LET palabra=8
1020 RESTORE
```

```
1030 LET elige=INT (RND*palabra)+1
1040 FOR l=1 TO elige
1050 READ p$
1060 NEXT l
1070 LET longitud=LEN p$: LET d$=p$
1080 RETURN
2000 REM titulo
2010 CLS
2020 FOR l=5 TO 10: GO SUB 7000+l*100: NEXT l
2030 PAPER 7: INK 0: PRINT AT 4,2;"H O M B R E  A H O R C A D O"
2040 PRINT AT 21,5;"pulsa ENTER para empezar": INPUT a$
2045 CLS
2050 PRINT AT 2,2;"Letras escogidas"
2060 PRINT AT 3,2;"----------------"
2070 PRINT AT 12,2;"La palabra"
2080 PRINT AT 14,2;"del ordenador es"
2090 INK 2: PRINT AT 17,2;
2100 FOR l=1 TO longitud: PRINT "*";: NEXT l
2110 INK 0: PRINT AT 20,2;"Por favor, escoge"
2120 PRINT AT 21,6;"una letra"
2130 LET ganas=0: LET pierdes=0: LET vida=0: LET puntos=0
2140 RETURN
3000 REM elige letra
3010 INPUT c$
3015 BEEP .1,20
3020 PAPER 7: INK 0: PRINT AT 5,vida;c$
3030 IF c$>CHR$ 96 THEN  LET c$=CHR$ (CODE c$-32)
3040 LET correcto=0
3050 FOR p=1 TO longitud
3060 IF c$<>p$(p) THEN  GO TO 3110
3070 LET correcto=1: LET puntos=puntos+1
3080 LET p$=p$(1 TO p-1)+"?"+p$(p+1 TO longitud)
3090 INK 2: PRINT AT 17,p+1;c$
3110 NEXT p
3120 IF puntos=longitud  THEN  LET ganas=1
3130 RETURN
4000 REM ganar
4010 CLS
4020 FOR l=5 TO 10: GO SUB 7000+l*100: NEXT l
4030 INK 0: PAPER 7: PRINT AT 1,4;"Bien hecho ! Tu ejecucion"
4040 PRINT AT 2,4;"queda aplazada"
*4045 PLOT 222,110: DRAW -3,3
*4046 PLOT 232,110: DRAW 3,3
*4047 FOR s=0 TO 40: BEEP .05,s: NEXT s
4050 RETURN
5000 REM perder
5010 PAPER 7: INK 0: CLS
5020 FOR l=1 TO 10: GO SUB 7000+l*100: NEXT l
5030 INK 2: PAPER 7: PRINT AT 7,2;"Inutil !"
5040 PRINT AT 12,2;"Has perdido !"
5050 PRINT AT 14,2;"La palabra era ";: PRINT AT 16,5;d$
*5051 FOR s=40 TO 0 STEP -1: BEEP .05,s: NEXT s
*5052 PRINT AT 6,27;"- -"
*5053 PLOT 222,110: DRAW -3,-3
*5054 PLOT 232,110: DRAW 3,-3
*5055 PRINT AT 20,26;" ": PRINT AT 20,30;" "
*5056 PAPER 0: PRINT AT 21,27;" ": PRINT AT 21,29;" ": PAPER 7
5060 RETURN
6000 REM otra partida
6010 PRINT AT 20,1;"Quieres jugar"
6020 PRINT AT 21,1;"otra partida ?"
6030 INPUT a$
6040 IF a$(1)="s" OR a$(1)="S" THEN  GO TO 10
6050 CLS : PAPER 7: INK 2: PRINT AT 10,10;"A D I O S"
6060 RETURN
7000 REM hombre ahorcado
```

```
7010 LET vida=vida+1: IF vida=10 THEN  LET pierdes=1
7020 GO TO 7000+vida*100
7100 REM patibulo 1
7110 INK 0
7120 FOR r=0 TO 21: FOR c=19 TO 20
7130 PRINT AT r,c;CHR$ 143
7140 NEXT c: NEXT r
7150 RETURN
7200 REM patibulo 2
7210 INK 0
7220 FOR r=0 TO 1: FOR c=21 TO 30
7230 PRINT AT r,c;CHR$ 143
7240 NEXT c: NEXT r
7250 RETURN
7300 REM patibulo 3
7310 INK 0
7320 FOR r=100 TO 105
7330 PLOT 155,r: DRAW 60,60
7340 NEXT r
7400 REM cuerda
7410 INK 0: PLOT 227,135: DRAW 0,25
7420 RETURN
7500 REM cara
7510 INK 0: CIRCLE 227,119,15
7520 PRINT AT 6,27;"+ +"
7530 PLOT 227,115: DRAW 0,7
7540 PLOT 222,110: DRAW 10,0
7550 RETURN
7600 REM cuerpo
7610 PAPER 2
7620 FOR r=9 TO 13: PRINT AT r,27;"   ": NEXT r
7630>INK 7:PRINT AT 10,28;"O":PRINT AT 12,28;"O"
7640 RETURN
7700 REM brazo 1
7710 PAPER 2: INK 7
7720 FOR r=9 TO 14: PRINT AT r,26;" ": NEXT r
7730 PLOT 215,56: DRAW 0,35
7740 PAPER 7: INK 0: CIRCLE 210,50,4
7750 RETURN
7800 REM brazo 2
7810 PAPER 2: INK 7
7820 FOR r=9 TO 14: PRINT AT r,30;" ": NEXT r
7830 PLOT 240,56: DRAW 0,35
7840 PAPER 7: INK 0: CIRCLE 244,50,4
7850 RETURN
7900 REM pierna 1
7910 PAPER 6: INK 7
7920 FOR r=14 TO 19: PRINT AT r,27;CHR$ 128;CHR$ 133: NEXT r
7930 RETURN
8000 REM pierna 2
8010 PAPER 6
8020 FOR r=14 TO 19: PRINT AT r,28;"  ": NEXT r
8030 REM cinturon y zapatos
8040 INK 0
8050 FOR c=60 TO 63
8060 PLOT 216,c: DRAW 23,0
8070 NEXT c
8080 INK 7: CIRCLE 228,61,3
8090 PAPER 0: INK 7
8100 PRINT AT 20,26;"  ": PRINT AT 20,29;"  "
8110 PLOT 228,16: DRAW 0,35
8120 REM retardo
8130 FOR d=1 TO 100: NEXT d
8140 RETURN
9000 DATA "MANZANA","ANGULO","ALFABETO","ANTIGUO"
9010 DATA "ALUBIAS","LIBRO","BUFALO","PLAYA"
```


ESTA NUEVA SERIE CONSTITUYE UNA INTRODUCCIÓN A
EL ARTE DE PROGRAMAR ORDENADORES. CADA LIBRO
ENSEÑA CÓMO ESTRUCTURAR UN PROGRAMA EN
RUTINAS, Y AL MISMO TIEMPO EXPLICA Y
ANALIZA LO QUE SE LE PIDE AL ORDENADOR
QUE HAGA Y POR QUÉ.
¡Y ADEMÁS ES DIVERTIDO! CADA LIBRO CONTIENE
COMO MÍNIMO UN JUEGO DE ORDENADOR,
APASIONANTE Y ORIGINAL, ESTRUCTURADO
DE TAL MODO QUE PUEDE AMPLIARSE
O REDUCIRSE, DE ACUERDO CON EL NIVEL
O LOS DESEOS DEL USUARIO.
TÍTULOS DE LA SERIE
Gráficos - Iniciación al basic
El banco de datos